

Universidade do Minho
Instituto de Educação

Nuno Miguel Machado Afonso

**Da tarefa ao projeto – Uma visão
construtivista do ensino da Programação
Orientada a Objetos**



Universidade do Minho
Instituto de Educação

Nuno Miguel Machado Afonso

**Da tarefa ao projeto – Uma visão
construtivista do ensino da Programação
Orientada a Objetos**

Relatório de Estágio
Mestrado em Ensino de Informática

Trabalho realizado sob orientação do
Professor Doutor Bento Duarte da Silva

setembro de 2013

DECLARAÇÃO

Nome: Nuno Miguel Machado Afonso

Endereço eletrónico: afonsonuno@sapo.pt

Número do Bilhete de Identidade: 11670902

Título do Relatório: “Da tarefa ao projeto – Uma visão construtivista do ensino da Programação Orientada a Objetos”

Orientador: Professor Doutor Bento Duarte da Silva

Ano de conclusão: 2013

Designação do Mestrado: Mestrado em Ensino de Informática

É AUTORIZADA A REPRODUÇÃO PARCIAL DESTE RELATÓRIO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, ____/____/____

Assinatura:_____

AGRADECIMENTOS

Aos meus pais e à restante família pela ajuda que me deram ao longo deste percurso académico.

A todos os meus amigos que, de alguma forma, me auxiliaram nesta etapa da minha vida.

Ao professor Doutor Bento Duarte da Silva, supervisor do estágio, por toda a colaboração, orientação e tempo despendido ao longo deste projeto.

Uma palavra de apreço ao professor Hélio Vilas, orientador cooperante do estágio, pela partilha de conhecimento e por ter sempre demonstrado grande disponibilidade para ajudar.

Aos alunos do turno 2 da turma 2P do curso Profissional de Técnico de Gestão e Programação de Sistemas Informáticos, pela participação e empenho que demonstraram ao longo de todo o projeto e aos quais desejo o melhor no seu futuro pessoal e profissional.

RESUMO

“Da tarefa ao projeto – Uma visão construtivista do ensino da Programação Orientada a Objetos”

Este estudo teve como objetivos perceber a contribuição de uma abordagem construtivista no processo de ensino-aprendizagem da Programação Orientada a Objetos num curso profissional de Técnico de Gestão e Programação de Sistemas Informáticos e compreender até que ponto esta abordagem contribui para fomentar a autonomia e a motivação dos alunos. Durante esta investigação procurou-se estabelecer um equilíbrio entre a tecnologia, a metodologia e as características e necessidades de aprendizagem dos alunos concebendo-se assim um ambiente concretizador de aprendizagens. Este ambiente permitiu melhorias significativas num conjunto de alunos que à partida demonstravam um quadro de desinteresse pela aprendizagem, problemas de comportamento e autonomia.

Tendo por base as metodologias *stepwise improvement* e *worked example* utilizadas no ensino da programação na Dinamarca e metodologias baseadas em princípios construtivistas, inclusive a teoria da flexibilidade cognitiva e aprendizagem baseada em problemas e projetos, desenvolveu-se um conjunto de materiais didáticos e estratégias, que permitiram aos alunos, tarefa a tarefa, concretizarem com sucesso as aprendizagens pretendidas. Tendo em vista o sucesso desta concretização procurou-se durante este processo promover o gosto pelo aprender e fazer através da diversificação de estratégias e da participação ativa dos alunos. Os resultados do estudo mostram que a metodologia aplicada é uma boa prática no ensino da programação nos cursos de formação profissional.

ABSTRACT

"From task to the project - a constructivist view of teaching Object-oriented Programming"

The objective of this study is understanding the contribution of the constructivist approach in the learning process of object-oriented programming in a professional course of Technical management and programming systems and realize how this approach contributes to improve the students autonomy and motivation. During this investigation, a balance between technology, methodology and the students' characteristics and learning needs was established to conceive a concretizing environment of learning. This environment allowed significant improvements in a students' group that showed lack of interest, autonomy and behavior problems.

Based on Stepwise improvement and Worked example methodologies used in the teaching of programming language in Denmark and methodologies based in constructivist approach, including problem-based learning, project-based learning and cognitive flexibility, have developed a set of teaching materials and strategies that allowed the students to successfully achieve the intended learning step by step was developed. Aiming the success, during this process, the pleasure of learning and doing, through diversification strategies and active participation of students is highly promoted. The study results show that the methodology used is quite effective in the teaching of programming in vocational courses.

ÍNDICE GERAL

INTRODUÇÃO.....	13
CAPÍTULO I – CONTEXTUALIZAÇÃO DA INTERVENÇÃO.....	17
1.1 O Construtivismo	17
1.1.1 Aprendizagem baseada em problemas.....	20
1.1.2 Aprendizagem baseada em projetos	21
1.1.3 Teoria da flexibilidade cognitiva	21
1.2 Metodologias de ensino da programação na Dinamarca	23
1.3 A autonomia na aprendizagem.....	27
1.4 A motivação para a aprendizagem	28
1.5 O contexto de intervenção	29
1.5.1 A escola.....	29
1.5.2 A turma.....	31
1.6 O plano geral de intervenção	32
CAPÍTULO II – DESENVOLVIMENTO E AVALIAÇÃO DA INTERVENÇÃO.....	37
2.1 A concepção dos ambientes de aprendizagem.....	37
2.2 O projeto desenvolvido pelos alunos ao longo da intervenção	40
2.3 Tecnologia educativa	44
2.3.1 Prezi.....	44
2.3.2 Plataforma moodle.....	45
2.3.3 Microsoft Visual C# 2010	46
2.3.4 Dropbox.....	49
2.4 Atividades e estratégias realizadas no âmbito do projeto	50
2.4.1 Aulas 1 e 2	53
2.4.2 Análise, compreensão e avaliação da ação.....	66
2.4.3 Aulas 3 e 4	67
2.4.4 Análise, compreensão e avaliação da ação.....	70
2.4.5 Aulas 5 e 6	71
2.4.6 Análise, compreensão e avaliação da ação.....	72
2.4.7 Aulas 7 e 8	72
2.4.8 Análise, compreensão e avaliação da ação.....	78
2.4.9 Aulas 9 e 10	79
2.4.10 Análise, compreensão e avaliação da ação.....	80

2.4.11	Aulas 11 e 12	81
2.4.12	Análise, compreensão e avaliação da ação	83
2.4.13	Aulas 13 e 14	84
2.4.14	Análise, compreensão e avaliação da ação	84
2.4.15	Aulas 15 e 16	84
2.4.16	Análise, compreensão e avaliação da ação	85
2.4.17	Aulas 17 e 18	85
2.4.18	Análise, compreensão e avaliação da ação	86
2.5	Avaliação do Projeto de intervenção	86
2.5.3	Avaliação dos projetos desenvolvidos pelos alunos	86
2.5.4	Análise da progressão da autonomia, motivação e comportamento	87
2.5.5	A opinião dos alunos	89
2.5.6	Avaliação do projeto de intervenção à luz dos objetivos	91
2.6	Dificuldades encontradas	95
CONCLUSÕES		97
BALANÇO FINAL DA FORMAÇÃO		101
REFERÊNCIAS BIBLIOGRÁFICAS		105
ANEXOS		109
Anexo 1: Planificação do módulo 11		109
Anexo 2: Guião da tarefa 1		111
Anexo 3: Guião da tarefa 2		113
Anexo 4: Guião da tarefa 3		117
Anexo 5: Guião da tarefa 4		119
Anexo 6: Diário de aprendizagem		121
Anexo 7: Grelha de avaliação do projeto desenvolvido pelos alunos		123
Anexo 8: Grelha de observação de comportamentos		125
Anexo 9: Questionário de autoavaliação dos alunos e avaliação do projeto de intervenção pedagógica.		127
Anexo 10: Exemplo de projeto desenvolvido pelos alunos		129
Anexo 11: Relatório de visualizações por material na plataforma moodle.		135

ÍNDICE DE FIGURAS

Figura 1 - Metodologia stepwise improvement.....	25
Figura 2 - Metodologia worked example	26
Figura 3 - Conceção de um ambiente facilitador de aprendizagens.....	38
Figura 4 - Conceção de um ambiente de aprendizagem que promova a participação ativa.....	39
Figura 5 - Divisão em tarefas do projeto a ser desenvolvido pelos alunos	41
Figura 6 - Esquema exemplificativo da relação entre as aulas.....	42
Figura 7 - Template do prezi escolhido para a intervenção.....	44
Figura 8 - Organização da plataforma moodle.....	45
Figura 9 - Feedback da tecnologia Microsoft Visual C# 2010	48
Figura10 - Momento do prezi que fomentou o brainstorming.....	53
Figura11 - Esquema demonstrativo do funcionamento das streams	55
Figura12 - Analogia como facilitadora da aprendizagem de Streams de input e output.....	55
Figura13 - Introdução dos conceitos de filestream e memorystream	57
Figura14 - Processo de carregar um ficheiro para memória RAM	57
Figura15 - Código de leitura de um ficheiro para memória RAM	58
Figura16 - Momento do prezi em que é solicitado aos alunos que realizem a atividade.....	58
Figura17 - Exemplo de funcionamento da classe bufferedStream	59
Figura18 - Analogia com o ciclo da água.....	60
Figura19 - Problema do hotel.....	62
Figura 20 - Prezi de demonstração do projeto.....	63
Figura 21 - Mapa mental 1 utilizado nas aulas 3 e 4	67
Figura 22 - Mapa mental 2 utilizado no final das aulas	68
Figura 23 - Problema “nomes e apelidos”	69
Figura 24 - Problema das anedotas	71
Figura 25 - Hierarquia de streams	73
Figura 26 - Conceito de coleção.....	74
Figura 27 - Código apresentado aos alunos para comentarem	75
Figura 28 - Exemplo de código comentado pelos alunos.....	75
Figura 29 - Atividade colaborativa do problema do Call Center.	76
Figura 30 - Mapa mental para relacionar a aula com os assuntos precedentes	79
Figura 31 - Mapa mental para relacionar conteúdos	81
Figura 32 - Analogia entre a serialização e o sapo insuflável	82
Figura 33 - Problema sobre serialização.	83
Figura 34 - Esquematização desde a conceção do projeto ao seu feedback	84
Figura 35 - Esquematização desde a conceção do projeto à sua avaliação	85
Figura 36 - Esquematização da partilha de dados com professor responsável pelo outro turno	96

ÍNDICE DE TABELAS

Tabela 1 – Calendarização e distribuição das atividades.....	52
Tabela 2 – Lista de temas escolhidos pelos alunos.....	64
Tabela 3 – Avaliação individual do projeto desenvolvido pelos alunos.....	87
Tabela 4 – Indicadores do desempenho dos alunos	88

ÍNDICE DOS GRÁFICOS

Gráfico 1 – Síntese dos dados recolhidos relativamente à autonomia, motivação e comportamentos..	88
Gráficos 2,3 e 4 – Opinião dos alunos sobre o projeto de intervenção.....	89

INTRODUÇÃO

Este relatório é resultado do projeto de intervenção pedagógica supervisionada, realizado no âmbito da unidade curricular Estágio Profissional do 2º ano do ciclo de estudos conducente ao grau de mestre em Ensino de Informática e descreve de forma fiel e transparente todo o processo de investigação-ação.

O tema deste projeto comporta a utilização de uma abordagem construtivista no ensino da Programação Orientada a Objetos, trabalhando sobre a tecnologia *Microsoft Visual Studio C# 2010*, como estratégia para promover a motivação, a autonomia e a responsabilidade dos alunos. Nesta perspetiva de abertura à mudança, procurou-se que os alunos também adquirissem técnicas de programação e desenvolvimento de sistemas informáticos, através de metodologias baseadas no paradigma construtivista como a aprendizagem baseada em problemas, aprendizagem baseada em projetos, a teoria da flexibilidade cognitiva e o uso de analogias.

Este tema é bastante pertinente, na medida em que o público-alvo deste projeto pertence ao universo dos cursos profissionais, e estando estes direcionados para a formação de técnicos aptos a ingressarem no mercado de trabalho, procurou-se proporcionar ambientes de aprendizagem que promovessem a formação para o exercício de uma profissão. São inerentes a esta formação capacidades de trabalhar autonomamente e com motivação, resolvendo problemas e projetos próximos do real.

Assim sendo, uma abordagem construtivista permite que os alunos construam as aprendizagens a partir dos seus conhecimentos prévios, estabelecendo analogias, desenvolvendo projetos e resolvendo problemas que simulem casos reais. Todo este processo de ensino-aprendizagem torna os jovens mais criativos e independentes, dotados de uma aprendizagem para o uso. Contrapõem-se assim metodologias mais tradicionais de ensino da programação, onde os alunos passivamente decoram conteúdos e exemplos, apresentando depois graves dificuldades em conseguir compreender novas situações e aplicar o conhecimento adquirido a novos contextos. É então necessário que os professores adotem um papel, mais de supervisores e orientadores do processo de ensino-aprendizagem e permitam aos alunos assumir o papel ativo neste processo.

A abordagem seguida no desenvolvimento do projeto assenta em pressupostos construtivistas, e foi complementada por algumas metodologias do ensino da programação na Dinamarca, como a *stepwise improvement* e *worked example*.

De acordo com os dados recolhidos, durante o período de observação das aulas lecionadas pelo meu orientador e através dos elementos recolhidos através da análise documental, nomeadamente dossiê de turma, atas anteriores à intervenção, listagem de faltas e módulos em atraso, foi possível perceber a pertinência e necessidade da mudança de comportamentos. Nesse sentido, foi desenvolvido este projeto de investigação-ação, e implementadas estratégias que, obedecendo às temáticas, permitiram aos alunos desenvolver competências científicas ao nível da Programação Orientada a Objetos, preparando-os para o uso das mesmas em situações reais. Em simultâneo procurou-se também promover a autonomia, reduzindo a dependência que os alunos tinham do professor, e a motivação, construindo um sentimento de autoeficácia. Todos estes elementos levaram a uma participação mais ativa dos alunos e por sua vez à melhoria de comportamentos.

Tendo por base estas intenções definiram-se como objetivos principais:

- (i) Avaliar o impacto da abordagem construtivista no ensino-aprendizagem da Programação Orientada a Objetos;
- (ii) Verificar a adequação de alguns princípios didáticos de ensino da informática na Dinamarca ao nosso sistema de ensino profissional;
- (iii) Analisar o contributo da abordagem construtivista e das metodologias de ensino da programação na Dinamarca na promoção da autonomia e motivação dos alunos.

Numa perspetiva reflexiva e apoiada na literatura, observou-se a evolução das competências evidenciadas pelos alunos e a alteração de comportamentos, para tal recorreu-se a estratégias de investigação pedagógica como suporte à compreensão e avaliação das práticas de ensino-aprendizagem, através da recolha de informação baseada em exercícios diagnósticos, diários de bordo, grelhas de observação de comportamentos e nos diários de aprendizagem preenchidos pelos alunos. Estes dados permitiram a definição e redefinição de estratégias pedagógicas, no sentido de encontrar práticas educativas que permitissem ultrapassar as dificuldades encontradas.

Este relatório é composto por dois capítulos. No capítulo I é apresentada inicialmente uma contextualização teórica do conceito de autonomia, motivação, construtivismo, das metodologias a ele associadas e das metodologias de ensino da informática na Dinamarca utilizadas no âmbito deste projeto. De seguida é contextualizada a escola, a turma e apresentado o Plano Geral de Intervenção: objetivos; estratégias; investigação e avaliação da ação. No capítulo II é apresentado, de forma detalhada, o trabalho realizado ao longo do projeto de intervenção pedagógica, são descritas as

atividades, refletindo-se sobre a criação das mesmas. Também neste capítulo são apresentados os recursos educativos utilizados, resultados gerais e dificuldades encontradas.

Nas considerações finais apresentam-se as conclusões e limitações do projeto, apontando recomendações didáticas e de investigação deles emergentes. É também apresentado um balanço sobre o valor do projeto no desenvolvimento pessoal e profissional.

CAPÍTULO I – CONTEXTUALIZAÇÃO DA INTERVENÇÃO

1.1 O Construtivismo

“Se deres um peixe a um homem, ele alimentar-se-á uma vez. Se o ensinares a pescar, alimentar-se-á durante toda a vida”

Kuan-Tsu

A abordagem construtivista surge a partir dos anos 80 como uma alternativa ao ensino dito tradicional, no qual o aluno assume um papel passivo no processo de ensino-aprendizagem, como mero espectador, e o professor age como agente ativo de todo o processo expondo a matéria a memorizar. Este processo de decorar conteúdos e exemplos impede o aluno de adquirir a flexibilidade cognitiva imprescindível para adaptar os conhecimentos adquiridos a novas situações.

Para este paradigma construtivista muito contribuiu Piaget (1979), ao referir que a criança quando nasce, apesar da sua bagagem hereditária, não consegue realizar a mais simples operação de pensamento e, como tal, vai ter de passar por um processo de construção e reorganização progressiva.

Esta construção ocorre sempre que uma criança se depara com uma nova situação com a qual não está familiarizada. Esta nova situação cria um estado de desequilíbrio na sua mente que necessita de ser mentalmente resolvido. Para se equilibrar, a nível cognitivo, a criança tem de modificar ou reestruturar o seu esquema mental para compreender esta nova situação. Os processos mentais responsáveis por esta reestruturação do esquema são a assimilação e a acomodação, “dois polos de uma interação que se desenvolve entre o organismo e o meio, a qual constitui a condição indispensável de todo o funcionamento biológico e intelectual” (Piaget, 1979, p. 328). Assim sendo, a integração de novas informações com o conhecimento existente é um processo cognitivo que Piaget designa por assimilação, ou seja, quando as crianças encontram novas situações de aprendizagem utilizam os conhecimentos prévios para que as novas experiências se tornem compreensíveis. Na assimilação a criança não modifica as suas estruturas, simplesmente incorpora a nova informação.

Por fim, este conhecimento prévio é reestruturado para que a nova experiência se acomode ao novo esquema, ou seja, o esquema inicial transforma-se em função dos novos objetos ou do próprio meio. A inteligência está na capacidade de coordenar a assimilação e a acomodação, e esta “evolui de um estado em que a acomodação ao meio é indiferenciada da assimilação para um estado onde se torna distinta da assimilação” (idem). Com o crescimento, a criança passa de uma fase onde assimila

o meio, para uma fase mais complexa, onde se multiplicam as acomodações devido às exigências do meio, e há maior necessidade de relacionamento e coordenação entre esquemas, isso acontece porque "o sujeito interessa-se pela novidade e ele próprio a explora, de facto quanto mais os esquemas [assimilação e acomodação] se distanciam mais diminui a defasagem entre o novo e o conhecido" (idem, p. 329). A novidade é vista como um problema que solicita a curiosidade, o desafio, a vontade de explorar e pesquisar.

Tendo por base estes processos que definem a interação adaptativa do sujeito ao meio, a aprendizagem é vista aos olhos do paradigma construtivista como um processo ativo e construtivo, onde o aluno é um construtor de conhecimentos. Os conhecimentos são construídos através da contextualização, com base em experiências pessoais e conhecimentos prévios. Onde os alunos aprendem ajustando a nova informação à anteriormente existente.

Este papel mais ativo, resulta num maior envolvimento dos alunos e num maior gosto pelas aprendizagens.

Como refere Ausubel (1978) o aprendiz é o artifice, o verdadeiro ator do processo de aprendizagem, uma vez que as novas aprendizagens só serão possíveis a partir dos conceitos, crenças, representações, conhecimentos e destrezas que este construiu no decorrer das suas experiências prévias, ou seja "é algo que ninguém pode realizar em seu lugar" (Coll et al., 2001, p. 85).

Segundo esta abordagem, o aluno é responsável pela construção das representações internas do seu conhecimento, baseando-se na participação ativa em experiências. Esta construção fica sempre em aberto para que novas associações e estruturações permitam a incorporação de novos conhecimentos.

Todo este conceito desvia-se do ensino tradicional, que obriga o aluno a repetir inúmeras vezes os conceitos teóricos até à sua memorização, vendo-o como uma folha de papel em branco, sem experiências e conhecimentos prévios. Aos olhos dos construtivistas "o espírito não é uma tabula rasa sobre a qual se inscreveriam as vinculações já prontas e impostas pelo meio exterior" (Piaget, 1969, p. 48), mas sim algo que o próprio indivíduo tem de construir e reconstruir na sua interação com o meio onde opera. A linguagem que é o instrumento mais poderoso do ensino tradicional deve ser substituído pelas ações do aluno, as quais "organizadas enquanto esquemas de assimilação, possibilitam classificar, estabelecer relações, na ausência das quais aquilo que, por exemplo, se fala ou escreve perde seu sentido" (Macedo, 1994, pp. 15).

Neste contexto construtivista, para que os alunos assumam o protagonismo, o professor deve delegar-lhes o papel principal do processo de aprendizagem, passando a atuar "como guia e mediador

entre a criança e a cultura” (Coll et al., 2001, p. 23). Adotando um papel de orientador deve conduzir os diálogos e discussões e ser um facilitador de aprendizagens, através das quais os alunos sejam capazes de construir o seu conhecimento e promover a sua autonomia.

O professor ao implementar a abordagem construtivista na sua sala de aula deve tomar “decisões inteligentes inerentes à planificação, desenvolvimento e avaliação do ensino” (idem, p. 9), proporcionando aos alunos um rol de atividades e estratégias baseadas em projetos, na resolução de problemas, na aprendizagem pela descoberta, no raciocínio, no diálogo, na cooperação, na colaboração e no uso de analogias. Ou seja, deve proporcionar aos alunos ambientes de aprendizagem que ofereçam ferramentas de construção do conhecimento e a possibilidade de interação com a realidade.

A intervenção educativa deve então ser orientada para que o aluno desenvolva as capacidades de aprender a aprender, e aprender a fazer desenvolvendo a capacidade de aprender por si próprio realizando aprendizagens significativas.

Piaget (1969) defende que “a dificuldade maior na pedagogia (...) é de facto que os melhores métodos são também os mais difíceis” (p. 69). Isto acontece com a abordagem construtivista, pois carece de um grande esforço de bastidores, na preparação das aulas, na conceção de atividades que fomentem a criatividade e iniciativa, isto porque “o modo pelo qual o professor estrutura uma tarefa determina o grau com que os alunos podem ser autónomos e mostrar iniciativa” (Brooks & Brooks, 1997, p. 70).

Segundo Jonassen (1991), os alunos constroem a sua própria realidade com base nas suas interpretações das experiências e os professores devem-se concentrar em abordagens de ensino realistas que permitam o contacto com essas experiências através da resolução de problemas (*problem-based learning*) do mundo real.

1.1.1 Aprendizagem baseada em problemas

A aprendizagem baseada em problemas (*problem-based learning*), baseada nos pressupostos construtivistas, permite que os alunos construam as suas aprendizagens através do diagnóstico e resolução de problemas reais ou simulados. Segundo Cooper (1993), “para o construtivista, a aprendizagem é baseada na resolução de problemas através da descoberta pessoal, e o aprendiz está intrinsecamente motivado” (p. 13).

A resolução de problemas permite aos alunos transportarem as competências adquiridas num contexto para outro, pois “as virtudes reclamadas pela resolução de problemas não se resumem só à aquisição de conhecimento, mas também às habilidades de resolução de problemas que são retidas para uso em novos problemas, novas situações e novos contextos” (Watts, 1991, p. 7), assim sendo, os problemas apresentados devem ser o mais próximo possível daquilo que os alunos vão encontrar no seu mundo profissional.

Segundo Watts (1991), a abordagem baseada em problemas desenvolve-se a partir dos seguintes princípios sobre a aprendizagem, a referir: é um processo ativo, não passivo, e torna-se mais significativa se o aluno estiver totalmente envolvido no processo; é uma apropriação de competências e é para a vida.

Este processo decorre através da transferência de parte da responsabilidade da aprendizagem para os alunos, dando-lhes a oportunidade de tomarem decisões e pensarem por si mesmos. Assim, a resolução de problemas possibilita que sejam responsáveis pela organização do que estão a aprender, e que pratiquem e testem conceitos que já aprenderam. Sempre que as soluções vão resultando, os alunos vão ganhando confiança para atacarem futuros problemas que possam surgir.

Esta metodologia de ensino apresenta vantagens relevantes para a aprendizagem, entre as quais, a promoção da motivação, da criatividade, da autonomia, o desenvolvimento de capacidades de análise e a possibilidade de consolidação e interligação de saberes.

Watt (1991) refere um conjunto de razões pelas quais devemos optar pela resolução de problemas, por exemplo:

- Permite que os jovens se apropriem das tarefas.
- Encoraja a tomada de decisões.
- Permite uma aprendizagem ativa e por descoberta.
- É um veículo para a aprendizagem de várias competências científicas.

- Permite cruzar atividades curriculares.
- Fornece relevância e contextos da vida real (p. 15).

Segundo esta abordagem, os alunos desenvolvem métodos para construírem os seus próprios procedimentos, integrando o seu conhecimento conceitual com sua habilidade processual.

1.1.2 Aprendizagem baseada em projetos

Tal como na aprendizagem baseada em problemas, na aprendizagem baseada em projetos (*project-based learning*) o aluno assume o papel principal na construção do conhecimento. As duas abordagens são muito semelhantes e assentam nos mesmos princípios. Contudo diferem, essencialmente, no facto do projeto poder, ou não, passar pela resolução de um problema específico, pode ser simplesmente a programação de uma solução fornecida, e no facto dos projetos muitas vezes começarem com o produto final já em mente (Donnelly & Fitzmaurice, 2005).

Segundo Thomas (2000), esta metodologia de ensino acrescenta uma série de benefícios ao processo de ensino-aprendizagem na medida em que promove a participação, aumenta a autoconfiança e melhora as atitudes em relação à aprendizagem.

1.1.3 Teoria da flexibilidade cognitiva

Na programação o conhecimento não é rotineiro, mecanizado ou automatizado, pois para cada novo problema o conhecimento adquirido tem de ser adaptado à nova situação. Assim sendo, deve existir uma grande flexibilidade no seu controlo para que assim seja possível adaptá-lo a novas situações.

A teoria da flexibilidade cognitiva foi desenvolvida por Rand Spiro e colaboradores nos finais dos anos 80 e aplica-se a níveis avançados de aquisição de conhecimentos, em domínios complexos, caracterizando-se estes por um grande número de elementos e conteúdos que interagem entre si.

Segundo Spiro, Coulson, Feltovich e Anderson (1988), o controlo sobre as representações flexíveis dará ao indivíduo um melhor controlo sobre os conteúdos, e assim ao invés de estruturas monolíticas de conhecimento pré-embalado, que limitam o indivíduo a aplicá-lo de uma forma fixa e

limitada, o indivíduo pode controlar o conhecimento, ou seja, controlar uma grande variedade de formas para o adaptar a tarefas e conteúdos envolvidos numa nova situação.

Pretende-se com esta teoria criar práticas educacionais que permitam aos alunos aplicar melhor os conhecimentos que adquirem a novos casos do mundo real. Procura-se assim que o conhecimento seja construído para uso. Para tal, os alunos devem “alcançar uma compreensão profunda do conteúdo, raciocinar com ele, e aplicá-lo com flexibilidade em diversos contextos” (idem, p. 375), ao invés de reproduzi-lo simplesmente de forma imitativa como acontece nos modelos tradicionais de ensino.

Representações rígidas e pré-embaladas do conhecimento devem ser substituídas por representações flexíveis onde o conhecimento possa ser usado para satisfazer necessidades de determinados contextos. Esta construção de esquemas permite que o conhecimento seja utilizado de diferentes maneiras, em diferentes ocasiões para diferentes fins.

Aos olhos desta teoria não existe capacidade intelectual mais valorizada que a capacidade de um indivíduo pensar por si mesmo, tornando o conhecimento relevante através do treino num conjunto variado de situações. Neste processo o indivíduo adquire “a capacidade de usar a sua flexibilidade do conhecimento e eficientemente adaptá-lo a contextos variados” (Spiro, Vispoel, Schmitz, Samarapungavan & Boerger, 1987, p. 3).

Para se criar uma maior flexibilidade do conhecimento deve ser substituído o conhecimento demasiado compartimentado, por estruturas de conhecimento com um alto grau de interconexão. Em vez da simplificação devem ser tratados assuntos complexos. Este tipo de complexidade, numa situação final de aprendizagem, prepara os alunos para a complexidade de desafios com que se vão deparar no mundo real.

Segundo Spiro et al. (1987), a melhor maneira de aprender e ensinar, quando se quer desenvolver a flexibilidade cognitiva, para aplicação do conhecimento em situações futuras, é através de um método em que se trata o domínio dos conteúdos como uma paisagem que é explorada "cruzando" em muitas direções. Deste modo “aprende-se ao atravessar em várias direções as paisagens conceptuais” (Carvalho, 2011, p. 22).

Estes cruzamentos desenvolvem a flexibilidade com que o conhecimento é representado na memória, e o domínio que o indivíduo tem sobre a capacidade de articular os elementos das representações e reorganizar a importância dos mesmos. A qualidade e o desenvolvimento desta flexibilidade vão-se refletir no sucesso com que o conhecimento prévio é utilizado para a solução de uma nova situação.

Para que este processo decorra o professor deve “selecionar materiais de aprendizagem que proporcionem explorações multidimensionais da paisagem sob a ativa iniciativa do aluno” (idem).

1.2 Metodologias de ensino da programação na Dinamarca

A Dinamarca é reconhecida pela sua elevada qualidade de ensino, nomeadamente devido à prevalência da componente prática, flexibilidade dos seus programas curriculares, articulação com empresas e indústrias e à promoção da autonomia.

A Dinamarca tem 10 anos de escolaridade obrigatória¹, 1 ano de pré-escola e 9 anos de primário e secundário inferior, os alunos permanecem juntos durante todo o ensino obrigatório, recorrendo ao ensino diferenciado dentro da turma.

O último ano do pré-escolar (Børnehaveklasse) é obrigatório e faz a ponte entre a infância/creche e a escola primária. O jogo é um elemento importante desta fase de ensino e, como tal, abrange os seguintes temas: linguagem e formas de expressão, a natureza e os fenómenos das ciências naturais, música, movimento e função motora, habilidades sociais e relações interpessoais e de cooperação.

Concluído o pré-escolar, os alunos seguem para o primário e secundário inferior, durante esta fase a avaliação é contínua e formativa, tendo os professores que elaborar dois relatórios por ano sobre o progresso de cada aluno. Estes relatórios são disponibilizados aos encarregados de educação. A progressão para o nível seguinte é automática. Para conclusão destes níveis são disponibilizadas duas opções de exame, no 9º e/ou 10º ano. Ao longo deste percurso os alunos realizam uma série de testes com o objetivo de fornecer uma imagem precisa do nível académico de cada aluno.

Segundo Bang (2000), ao terminarem a escolaridade obrigatória, os alunos têm a possibilidade de frequentar um 10.º ano de escolaridade facultativo. Este ano opcional é frequentado geralmente por alunos que ainda não decidiram qual o futuro educativo/profissional que pretendem seguir ou que pretendem melhorar os seus conhecimentos gerais. Nesta fase os alunos podem prosseguir os seus estudos secundários, optando pela via do ensino geral, orientada para o acesso ao ensino superior, ou pela via do ensino profissional.

O ensino secundário superior da Dinamarca é muito diversificado e é oferecido por diversos tipos de instituições educativas.

¹ Informação disponível no portal do Ministério da Educação Dinamarquês em: <http://eng.uvm.dk/Education>

Os liceus, que permitem a obtenção do diploma de conclusão do secundário (Gymnasium), são instituições onde os alunos adquirem competências nas ciências humanas, naturais e sociais, com vista a estar em posição de continuar os estudos no Ensino Superior. São aplicados vários métodos de trabalho, tais como a instrução na sala de aula e a execução de projetos individuais e em grupo. Também costumam dedicar 25% do tempo ao ensino organizado, onde alunos e professores não estão, necessariamente, na mesma sala.

Os estabelecimentos específicos (HF), que também prepararam os jovens para a realização do exame de conclusão do secundário, são frequentados essencialmente por jovens e adultos que abandonaram e desejam retomar o sistema educativo.

As escolas comerciais que preparam os alunos para a obtenção do diploma superior de comércio (HHX), têm como objetivo proporcionar uma qualificação para estudos académicos superiores nas áreas da economia, empresariais e socioeconómicas.

As escolas técnicas, que permitem aos alunos obterem o diploma superior técnico (HTX), fornecem aos mesmos as competências no âmbito das ciências técnicas e naturais. Uma característica especial é permitir-lhes colocar em prática, o lado teórico do currículo em oficinas e laboratórios.

O ensino profissional que ocorre em escolas profissionais financiadas pelo Estado e consiste numa articulação do processo de aprendizagem entre as escolas profissionais e empresas privadas. Os alunos ganham competências para trabalhar numa profissão específica (eletricista, cozinheiro, mecânico), por isso, o ensino secundário profissional tem como objetivo a formação para o mercado de trabalho e não para o acesso ao ensino superior, embora seja possível, com certos requisitos, prosseguir os estudos a nível superior.

Descrito o funcionamento do sistema educativo Dinamarquês e direcionando-nos mais para o tema do projeto, ao nível do ensino da programação, segundo as orientações do MED (2010) ao nível didático, é aconselhável selecionar apenas uma linguagem de programação para uma boa organização educacional e clareza para os alunos.

Numa fase inicial, no ensino da programação devem-se utilizar programas simples, para que os alunos possam ler e explicar o seu comportamento. Num nível mais avançado, devem ser facultados programas que lhes permitam programar aplicações com interfaces gráficas.

Uma outra consideração didática passa por aplicar uma constante diversidade de estratégias e metodologias para manter a progressão dos alunos.

Uma das metodologias utilizadas no ensino da programação na Dinamarca e aplicada ao longo deste projeto foi a *stepwise improvement* (Melhoria gradual), que é uma abordagem didática e

metodológica para ensinar a programar. Consiste em pegar num projeto (problema) e subdividi-lo em várias partes, segundo Watts (1991), se um problema é demasiado grande, ao ponto de se tornar um obstáculo, é sempre possível dividi-lo em pequenas partes. O ensino é então organizado numa série de pequenas e gerenciáveis tarefas, alternando entre teoria/modelagem e implementação/teste para que os alunos experimentem uma correlação direta entre elas.

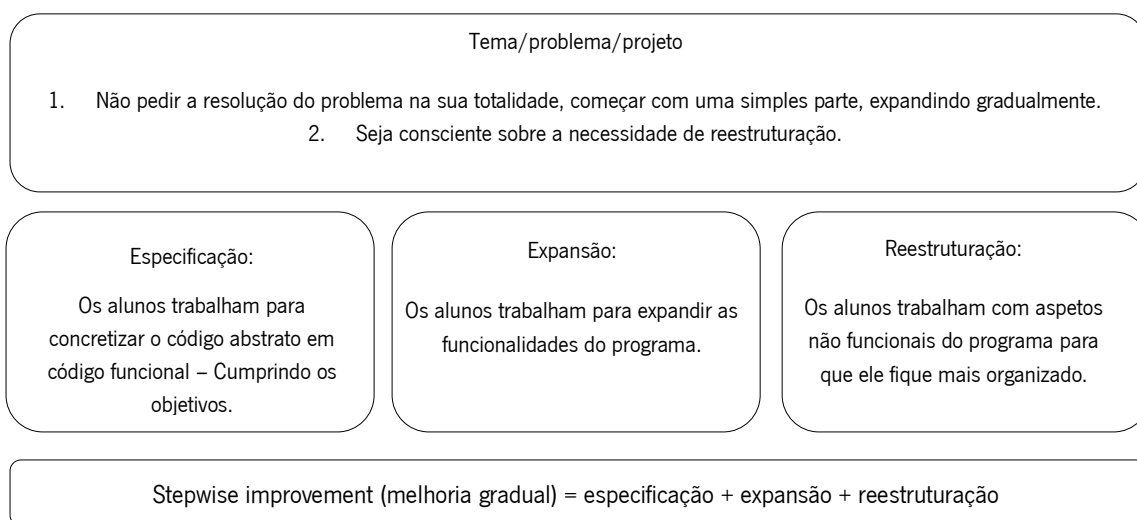


Figura 1 - Metodologia stepwise improvement.²

Esta metodologia permite aos alunos adquirirem progressivamente autonomia no processo de desenvolvimento do seu produto. O facto de sugerir a divisão do processo de desenvolvimento do projeto em várias etapas individuais permite aos alunos assumirem progressivamente, partindo do mais simples para o mais complexo, o processo de desenvolvimento dos seus trabalhos. Assim, tarefa a tarefa, os alunos vão ampliando as especificações do seu produto (expansão), aumentando assim as funcionalidades do mesmo.

Os alunos, com esta metodologia de melhoria gradual, podem mover-se em três dimensões diferentes: podem melhorar o seu projeto (corrigindo erros, por exemplo), podem expandi-lo (adicionando novas funcionalidades) ou reestruturá-lo (mudando a sua estrutura).

² Fonte: Ministério da Educação da Dinamarca

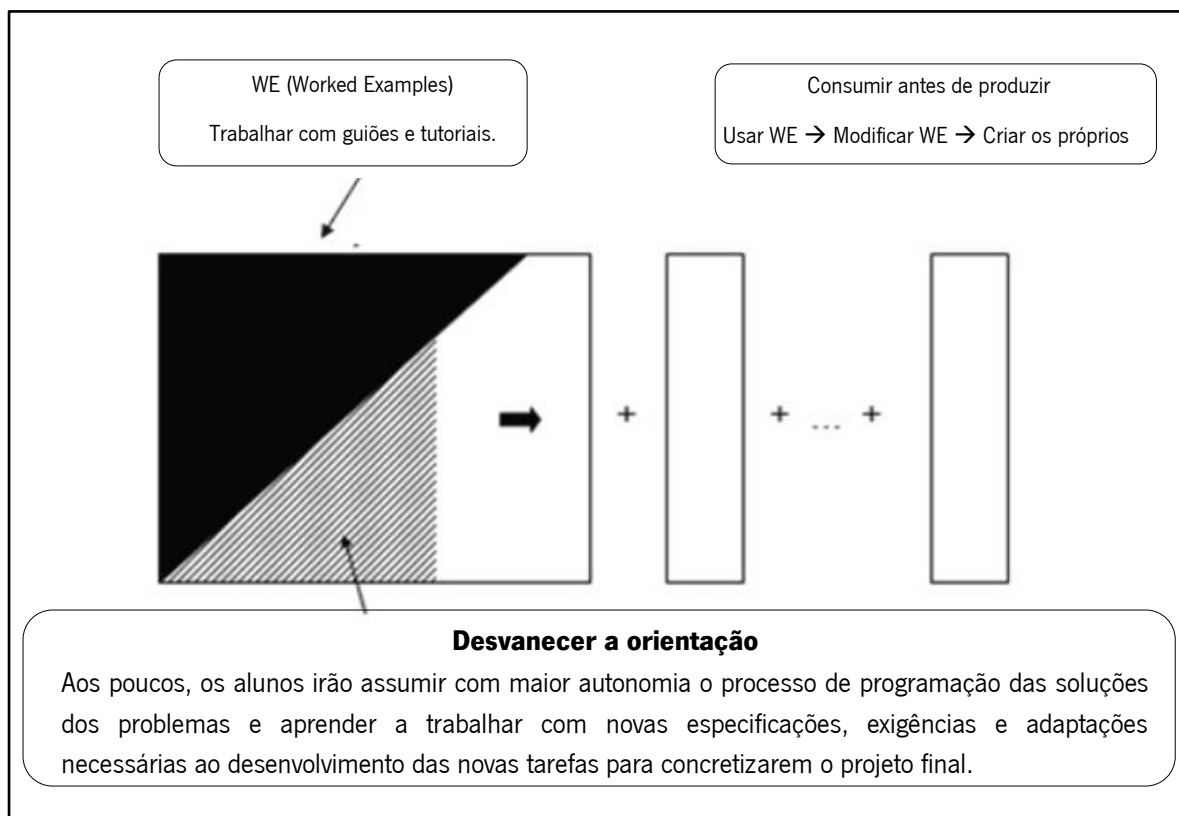


Figura 2 - Metodologia worked example

De acordo com MED (2010), recorrendo-se à metodologia *worked example*, alunos e professor começam a trabalhar a partir de uma situação inicial, com um elevado grau de controlo e ajuda do professor (área preenchida) através de exemplos, depois os alunos começam a ter de fazer escolhas e serem mais ativos na sua aprendizagem, trabalhando de uma forma mais independente (área sombreada), entretanto por fim, entram num processo de autoaprendizagem onde desenvolvem um produto autonomamente.

O ensino deve ser baseado em projetos através dos quais os alunos devem constantemente buscar novos conhecimentos. Estes novos conhecimentos devem ser integrados à experiência e ao conhecimento que o aluno já tem. A seleção do tema dos projetos e a organização do ensino deve ir de encontro ao interesse dos alunos a fim de despertar a curiosidade dos mesmos e motivar os seus esforços de aprendizagem. Também é sugerido que os alunos construam um portfólio com o material que vão recolhendo aula a aula, o que pode incluir desde reflexões até exercícios e projetos. Estes materiais podem ser guardados numa pasta ou num repositório online (de preferência uma ferramenta colaborativa).

1.3 A autonomia na aprendizagem

Considerando-se a autonomia como um elemento muito importante não só na aprendizagem da programação, mas em qualquer forma de aprendizagem, deve-se, enquanto professor, ter presente o objetivo de fomentar nos alunos a habilidade e capacidade de tomarem as suas próprias decisões. Nesta linha de pensamento, Dickinson (1987) refere que “um aprendiz autónomo é aquele que é totalmente responsável para tomar decisões que dizem respeito à sua aprendizagem” (p. 9). Esta oportunidade deve ser facultada aos alunos, como refere Freire (1997), construindo nos mesmos a capacidade e a liberdade de construírem e reconstruírem o que lhes é ensinado.

Embora Paulo Freire defenda que o aluno deve ser livre de construir as suas aprendizagens, não descarta o papel do professor que, na sua opinião, não é o de transmitir conhecimento, mas sim o de criar oportunidades para que os alunos de modo autónomo produzam e construam o seu conhecimento. Relativamente a este pensamento, verifica-se uma convergência de princípios entre a autonomia e o paradigma construtivista, pois ambos assumem como prioridade estimular o aluno a ser o elemento ativo da aprendizagem, respeitando os seus interesses e opiniões, promovendo assim a construção de inteligências ativas.

De acordo com Jiménez, Lamb e Vieira (2007), a conceção de autonomia do aluno assume-se como “a competência para se desenvolver como participante autodeterminado, socialmente responsável e criticamente consciente em (e para além de) ambientes educativos, por referência a uma visão da educação como espaço de emancipação (inter) pessoal e transformação social” (p. 27). O aluno assume-se como um ser crítico e reflexivo sobre as suas aprendizagens, capaz de construir competências que lhe permitam utilizar o conhecimento adquirido ao longo da sua vida.

1.4 A motivação para a aprendizagem

A motivação dos alunos tem implicações diretas na qualidade do seu empenho no processo de ensino-aprendizagem, pois o “aluno motivado busca novos conhecimentos e oportunidades, (...) envolve-se nas tarefas com entusiasmo e demonstra disposição para novos desafios” (Alcará & Guimarães, 2007, p. 177). Isto demonstra que a “motivação do aluno é uma variável relevante do processo ensino/aprendizagem, na medida em que o rendimento escolar não pode ser explicado unicamente por conceitos como inteligência, contexto familiar e condição socioeconómica” (Lourenço & Paiva, 2010, p. 133).

Podemos assim dizer que a motivação é a força que impulsiona o aluno na concretização das atividades. Segundo Boekaerts (2003), os alunos tentam dar sentido às novas situações de aprendizagem tendo por base as suas crenças motivacionais. Na base destas crenças está sempre uma necessidade, um desejo, uma intenção, um interesse, uma vontade ou uma predisposição para fazer algo.

Também nesta linha de pensamento, Atkinson (1964) define a motivação como aquela que impulsiona o comportamento e determina a direção, o vigor e a persistência da ação.

A motivação pode ser intrínseca ou extrínseca. Relativamente à motivação intrínseca, o aluno concretiza as tarefas, envolve-se com gosto no processo de aprendizagem, apenas pelo prazer de aprender, porque se interessa pelo assunto e obtém prazer simplesmente pela realização das atividades. No que diz respeito à motivação extrínseca, o aluno envolve-se no processo de aprendizagem por causas externas, como o reconhecimento do professor ou dos colegas, pela obtenção de recompensas, pelo medo de punições ou ainda pela necessidade da mesma para determinado fim.

O professor deve identificar a ausência de motivação dos alunos baseando-se em certos indicadores, tais como: “o fraco empenhamento do aluno nas tarefas escolares, a baixa participação dos alunos nas aulas e o pouco tempo despendido pelos alunos a estudar” (Jesus, 1997, p. 31). Da mesma forma que o professor deve ser capaz de detetar a falta de motivação, também deve ser capaz de conceber ambientes de aprendizagem que estimulem o aluno, despertem o seu interesse e lhe proporcionem satisfação. Outra vertente através da qual o professor pode fomentar a motivação dos alunos é através da criação de uma relação professor/aluno baseada em fatores efetivos, tais como a amizade, o respeito mútuo, a disponibilidade e a capacidade do professor entender o aluno.

1.5 O contexto de intervenção

1.5.1 A escola

O presente projeto de intervenção foi implementado na Escola Secundária de Alberto Sampaio (ESAS). A Escola fica situada na “freguesia de S. Lázaro da cidade de Braga” (IGEC, 2011, p. 2) e remonta a sua origem a 1884, então escola comercial de Braga. A população estudantil é muito diversificada, oriunda fundamentalmente, da zona centro e sul da cidade. A nível sociológico integra alunos provenientes não só de famílias de nível social médio-alto e alto, mas também uma parcela de alunos provenientes de famílias de nível médio-baixo e baixo, com baixos níveis de instrução. Na proximidade da escola “existem alguns dos bairros mais degradados da cidade, com problemas em matéria de segurança pública” (ESAS, 2011, p. 13).

Baseando-se numa comunidade educativa tão heterogénea, a escola apresenta uma oferta formativa diversificada, a referir: cursos científico-humanísticos; cursos profissionais; cursos de educação e formação de adultos; formações modulares; reconhecimento, validação e certificação de competências; e português para todos, orientado para estrangeiros.

A avaliação externa das escolas é responsabilidade da Inspeção Geral da Educação, e esta avaliação decorre da análise dos documentos fundamentais da escola, em especial da sua autoavaliação, dos indicadores de sucesso académico dos alunos, das respostas aos questionários de satisfação da comunidade e da realização de entrevistas. A ESAS foi avaliada com Muito Bom, tendo sido apontados os seguintes pontos fortes: a visão e a estratégia que, orientada por metas claras e avaliáveis, capacita a escola para pensar e reformular o seu futuro; as lideranças fortes e mobilizadoras de uma comunidade educativa profundamente identificada com a escola; e o desenvolvimento de uma cultura de avaliação bem enraizada, concretizada em práticas de autoavaliação consistentes e alargadas, determinante para o estabelecimento de planos de melhoria.

Como áreas que necessitam de ser melhoradas o relatório de avaliação externa aponta: as taxas de desistência escolar; a articulação com os estabelecimentos de proveniência dos alunos, de forma a garantir a sequencialidade das aprendizagens na transição do 9.º ano para o ensino secundário e a supervisão da prática letiva em sala de aula, a valorizar como estratégia de desenvolvimento e de estímulo à qualidade profissional e científica dos docentes.

As instalações escolares foram objeto de requalificação, no âmbito da intervenção do Parque Escolar, reunindo por isso as condições essenciais ao desenvolvimento da ação educativa das quais destacamos³:

- 42 Salas normais
- 8 Laboratórios Ciências Experimentais
- 2 Salas de Desenho
- 2 Salas de Oficina Artes
- 2 Laboratórios Multimédia
- 4 Laboratórios Software
- 4 Laboratórios Hardware
- 4 Auditórios
- 1 Sala de Música c/ estúdio para rádio e televisão
- 1 Pavilhão Desportivo
- 2 Ginásios
- 1 Bar Central
- 1 Sala de repouso (alunos) - polivalente
- 3 Salas de Estudo (alunos)
- 1 Cantina (2 pisos)
- 1 Biblioteca (2 pisos)
- 1 Sala Museu
- 1 Sala de Repouso com Bar - Professores
- 1 Sala de Departamentos - polivalente
- 2 Salas de reunião - Departamentos
- 2 Salas de reunião – Conselho Geral e Conselho Pedagógico
- 1 Sala de repouso – Pessoal não Docente
- Gabinete – Conselho Geral
- Gabinete da Associação de Pais
- Gabinete da Associação de Estudantes
- Gabinete de Apoio aos Alunos e Família e de Orientação Vocacional
- Gabinetes de Direção de Turma
- Gabinetes do Centro de Novas Oportunidades
- Gabinetes da Direção
- Outros Gabinetes – Áreas Específicas

³ Fonte Projeto Educativo ESAS 2011

1.5.2 A turma

Hoje em dia a maioria das escolas secundárias apresenta na sua oferta formativa os cursos profissionais, sendo uma das principais características desta opção de ensino “a ligação entre a educação, a formação e o mundo profissionalizante” (ANQEP, 2013, p. 5), ou seja, a aprendizagem realizada nestes cursos deve valorizar o desenvolvimento de competências para o exercício de uma profissão, em articulação com o setor empresarial local.

Os cursos profissionais são, então, um percurso alternativo que habilita os alunos com competências profissionais e pessoais para o exercício de uma profissão, prepara-os em simultâneo para, caso seja a sua vontade, seguirem para o ensino superior. São cursos que têm uma estrutura curricular distribuída em módulos, o que permite diferentes ritmos de aprendizagem, permitindo aos alunos auto proporem-se a fazer módulos em atraso sempre que se achem preparados para tal.

A nível curricular os cursos profissionais são agrupados em três componentes, a sociocultural, a científica e a técnica. Dentro da componente técnica encontra-se também a formação em contexto de trabalho, que complementa a formação profissional através de um período de estágio numa empresa.

No final do curso os alunos têm de realizar uma Prova de Aptidão Profissional (PAP), na qual demonstram as competências adquiridas ao longo da formação.

Este projeto de intervenção decorreu na turma do 11º ano do Curso Profissional de Técnico de Gestão e Programação de Sistemas Informáticos. Este curso visa formar técnicos em gestão e programação de sistemas informáticos, profissionais qualificados e aptos a realizar, de forma autónoma ou integrada numa equipa, atividades de conceção, especificação, projeto, implementação, avaliação, suporte, desenho e manutenção de sistemas informáticos.

A intervenção decorreu na disciplina de Programação e Sistemas de Informação, vulgarmente designada por PSI, que faz parte da componente técnica do curso, e é uma das disciplinas nucleares do curso tendo por isso a maior carga horaria, perfazendo esta 632 horas. Segundo a Direcção-Geral de Formação Vocacional (2005), esta disciplina tem por objetivo garantir aos alunos a aprendizagem de técnicas de programação e desenvolvimento de sistemas informáticos, necessárias ao sucesso profissional nesta área. Como esta é uma disciplina que pertence à componente técnica, a turma desdobra em dois turnos de 12 alunos cada, e o funcionamento em simultâneo dos mesmos, implica a existência de dois professores diferentes. Este desdobramento acontece para possibilitar ao professor um acompanhamento mais próximo dos alunos.

O projeto de intervenção terá lugar no segundo turno, a utilização deste resulta do facto de ser o único turno lecionado pelo professor orientador cooperante. Este turno é então constituído, como já foi referido, por 12 elementos, todos do sexo masculino. Com base na análise feita ao dossiê da turma: a idade média dos alunos é 17,6.

1.6 O plano geral de intervenção

No início do meu estágio profissional na escola, em setembro de 2012, após conhecer o meu orientador cooperante, comecei a assistir às aulas e a recolher informações sobre o contexto educativo e a turma, para assim desenvolver este plano geral de intervenção que serviu de base ao projeto descrito ao longo deste relatório.

Com base na análise na última ata de Conselho de Turma (2012), constatei que os alunos apresentavam problemas de comportamento, assiduidade, aproveitamento e motivação. O Conselho de Turma considerava que, na generalidade, os alunos eram faladores, distraídos, tinham falta de postura e concentração nas tarefas propostas. Estava registado em ata que alguns alunos perturbavam o bom funcionamento das aulas e tinham bastantes faltas às diferentes disciplinas. Os docentes corroboraram que, em termos gerais, os alunos apresentavam comportamentos disruptivos que perturbavam o bom funcionamento das aulas.

Após análise do dossiê da turma observei que:

- metade dos alunos (6 alunos) já tinha pelo menos uma retenção;
- a maioria tinha pais com baixas qualificações;
- alguns apresentavam famílias destruturadas;
- todos possuíam computador e tinham acesso à internet em casa.

Na mesma análise verifiquei que, quando responderam ao questionário inicial de caracterização, um dos alunos indicou a disciplina na qual decorreu esta intervenção como uma das quais tinha mais dificuldades e só dois a referiram como disciplina preferida.

De acordo com o documento fornecido pela Diretora de Turma, onde constavam todos os módulos em atraso à disciplina de Programação e Sistemas de Informação, pude constatar que mais de 50% dos alunos possuíam módulos em atraso à mesma.

Recorrendo à observação direta da dinâmica dos alunos em questão pude concluir que estes, na sua globalidade, eram pouco participativos, apresentavam problemas de comportamento, pouca motivação e fraca autonomia.

Toda esta análise e observação, associada ao facto da turma pertencer a um curso profissional, ao módulo lecionado durante esta intervenção ser o último da programação, com os conteúdos mais abstratos e complexos, levou-me a refletir e a procurar uma solução. Face a este conjunto de fatores, optei por utilizar uma abordagem pedagógica baseada no paradigma construtivista associado a metodologias do ensino da informática na Dinamarca. Esta opção surge por reconhecer as potencialidades destas abordagens.

Assim sendo, delineei os seguintes objetivos:

- Avaliar o impacto da abordagem construtivista no ensino da Programação Orientada a Objetos, no que respeita à promoção da autonomia e motivação.
- Verificar a adequação de alguns princípios didáticos de ensino da informática na Dinamarca ao nosso sistema de ensino.

Para a concretização destes objetivos foram delineadas algumas estratégias com o intuito de fomentar a aprendizagem significativa dos alunos.

Sendo assim, comecei por um aspeto fundamental enquanto professor, e enquanto defensor da abordagem construtivista, que reside no facto de ser crucial numa fase inicial o professor conhecer os seus alunos, os seus interesses e os seus conhecimentos prévios. Para isso, numa primeira fase, como já foi referido anteriormente, recolhi informações através do método de observação direta e da análise documental (como a planificação, Projeto Educativo da Escola, atas de reuniões, exercícios diagnósticos e dossier da turma). Numa segunda fase, planeei a aplicação de um exercício diagnóstico para aferir os conhecimentos prévios que os alunos têm da programação, pois segundo o paradigma construtivista, o professor deve estimular o aluno a aprender partindo dos conhecimentos prévios deste último, relacionando-os com novas situações, para que este possa, mais facilmente, atribuir um novo significado às novas aprendizagens.

Decidi então que deviam ser planeadas tarefas contextualizadas com a realidade e interesses dos alunos, com a intenção de cativar a sua atenção, permitir que obtenham aproveitamento e adquiram, de facto, competências técnicas, científicas e socioculturais, necessárias ao seu futuro profissional.

Por forma a permitir uma aprendizagem centrada na atividade realizada pelos alunos, criando um ambiente favorável ao desenvolvimento da autonomia e a aquisição de conhecimentos em

Programação Orientada a Objetos avançada, delinee para este projeto de intervenção a utilização de metodologias apoiadas na aprendizagem baseada em problemas e projetos.

Neste plano de intervenção propus o desenvolvimento de um projeto em grupos de 2 elementos, procurando promover assim o trabalho em grupo. Ao apostar na aprendizagem baseada em projetos, pretendi dar aos alunos a oportunidade de desenvolverem uma aplicação, na qual colocassem em prática os conteúdos da disciplina, mais propriamente o uso de *streams* e a leitura e escrita em ficheiros binários e texto. Perante os conteúdos a lecionar e as metodologias escolhidas, houve a necessidade escolher uma linguagem de programação e uma plataforma de suporte, após alguma reflexão, optei pela linguagem de programação C# usando o paradigma POO (Programação Orientada a Objetos) e a plataforma Visual Studio 2010 C# da Microsoft.

Relativamente à carga horária da disciplina, esta contemplava 6 blocos semanais de 90 minutos, sendo a sua distribuição feita da seguinte forma: 2 blocos na terça-feira, 2 blocos na quarta-feira e 2 blocos na quinta-feira. A implementação desta intervenção, relativamente a cada conjunto de dois blocos, foi desenhada com a seguinte distribuição, no primeiro bloco os primeiros 45 minutos deviam ser utilizados para exploração dos diversos conteúdos com os alunos (procurando estabelecer sempre a ligação entre os novos conceitos e os conceitos já existentes), os segundos 45 minutos para a realização de tarefas de consolidação de conteúdos, promovendo gradualmente a sua autonomia, e dando tempo e espaço aos alunos para construírem o seu conhecimento. Por fim, durante o segundo bloco seriam orientados e auxiliados na resolução do seu próprio projeto. Procurei que o desenho do projeto a ser desenvolvido pelos alunos fosse ao encontro dos seus interesses, no sentido de promover a sua motivação, uma vez que o desenvolvimento de projetos contribui para “promover a autonomia, a criatividade, a responsabilidade, ... na perspetiva de abertura à mudança, à diversidade cultural e ao exercício de uma cidadania ativa” (João, 2003, p. 5).

Decidi que após conclusão dos projetos, os alunos deviam apresentar à turma as diferentes soluções elaboradas, promovendo-se assim o desenvolvimento da oralidade, motivação e aprendizagem através da partilha de experiências e soluções.

Como é necessário existirem elementos de avaliação dos alunos, pois terá de ser apresentada no final uma nota correspondente ao desempenho de cada aluno e, do projeto de intervenção defini que o projeto desenvolvido pelos alunos seria objeto de avaliação. Assim, por forma a aferir de que modo o desenvolvimento do projeto pelos alunos contribuiu para a sua aprendizagem, decidi criar um instrumento de avaliação de acordo com os critérios definidos com a turma, enquadrados com os critérios de avaliação definidos pelo grupo disciplinar.

Também recorri a algumas estratégias e metodologias de ensino da informática na Dinamarca para planeamento deste projeto, as quais tomei conhecimento ao realizar um trabalho de pesquisa no âmbito da disciplina de Metodologias do Ensino da Informática II. Optei por estas estratégias, em primeiro lugar, porque é um país que promove a autonomia desde muito cedo, e ao longo de todas as fases de ensino, e em segundo lugar, porque achei que estas metodologias seriam parte da solução dos problemas com que me deparei.

Para que os alunos refletissem sobre o processo de aprendizagem ao longo do projeto e tivessem todos os seus materiais organizados, decidi sugerir a criação de um e-portfólio onde pudessem guardar todas as atividades realizadas e o respetivo projeto.

A fim de avaliar este projeto de intervenção e o cumprimento de seus objetivos pensei utilizar os seguintes instrumentos: observação direta como forma de compreender e analisar aspetos diversificados, nomeadamente o comportamento e envolvimento dos alunos na sala de aula; diários de aprendizagem a serem preenchidos pelos alunos no final de cada aula e grelhas de avaliação do comportamento em situação de aula. A avaliação do projeto em si foi planeada como contínua ao longo de todo o processo, permitindo assim a existência de espaço de reflexão e melhoria atempada de práticas.

Para fazer o levantamento da opinião final dos alunos sobre o projeto de intervenção, planeei elaborar um questionário *online* no *Google drive*, e disponibilizá-lo aos alunos através da plataforma moodle.

CAPÍTULO II – DESENVOLVIMENTO E AVALIAÇÃO DA INTERVENÇÃO

Neste capítulo, numa fase inicial serão apresentadas e discutidas algumas das opções que se tomaram relativamente a este projeto e a tecnologia utilizada. Numa fase mais avançada serão apresentadas, em moldes reflexivos, as atividades realizadas ao longo das aulas e alguns resultados a elas associados. Por fim, será apresentada a avaliação da intervenção.

2.1 A conceção dos ambientes de aprendizagem

Durante este projeto de investigação-ação procurei ajustar da melhor maneira possível, os princípios da abordagem construtivista aos alunos em questão, procurando assim promover o sucesso das suas aprendizagens. Para que isto fosse concretizável, acompanhei sempre os alunos de perto, refletindo sobre as dificuldades e problemas que foram surgindo, ajustando sempre a minha ação.

Tendo em atenção que estamos perante o ensino da Programação Orientada a Objetos no ensino profissional, para o sucesso deste processo de ensino-aprendizagem procurei proporcionar ambientes de aprendizagem onde existisse o equilíbrio, entre três elementos, na minha opinião, nucleares ao processo, a tecnologia, a metodologia (incluindo esta a postura do professor) e os alunos (tendo em conta as suas características, interesses e necessidades de aprendizagem). Pretendi com este equilíbrio que os alunos concretizassem aprendizagens significativas, digo significativas, porque pretendi que os alunos adquirissem competências realmente relevantes para o seu futuro profissional.

Depois de refletir sobre o assunto estabeleci que seria necessário, para este equilíbrio que todos os elementos chamados a participar neste processo de ensino-aprendizagem, realmente participassem e contribuíssem para o seu sucesso. Para que esta intenção fosse concretizável, conclui então, que os três elementos tinham de ter pontos em comum, ou seja se intersetassem, permitindo assim a criação de ambientes facilitadores de aprendizagens.

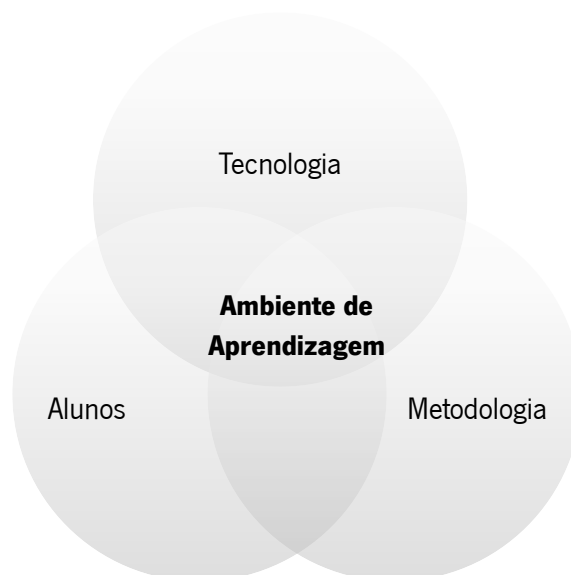


Figura 3 - Conceção de um ambiente facilitador de aprendizagens

Partindo sempre do elemento mais importante, o aluno, das suas necessidades de aprendizagem, motivações e interesses, optei por recorrer a tecnologias e metodologias, que partilhassem as mesmas características e propósitos com a intenção de atingir o sucesso das aprendizagens.

Seguidamente serão apresentados alguns exemplos que demonstram o cuidado prévio que se teve na escolha das metodologias e tecnologias.

Tendo em conta, que estamos na presença de alunos de um curso profissional, futuros técnicos, cujas aprendizagens devem ser feitas visando o seu futuro profissional, deve-se permitir que estes assumam o processo de aprendizagem. Tendo em atenção este ponto, procuraram-se tecnologias e metodologia que permitissem uma participação ativa dos alunos. Assim surgiu a tecnologia Microsoft Visual C#, que é uma aplicação que permite aos alunos construírem (programarem) ativamente as suas próprias soluções como se estivessem inseridos no mercado de trabalho. Da mesma forma, relativamente à metodologia, investigou-se e escolheram-se metodologias que permitissem uma participação ativa dos alunos, como a título exemplificativo a aprendizagem baseada na resolução de problemas e projetos, as quais relegam o professor para papel de orientador e facilitador. Como podemos verificar pela seguinte imagem (figura 4), utilizando a tecnologia Microsoft Visual C#, e metodologias baseadas em pressupostos construtivistas, estamos a criar um ambiente de aprendizagem que à partida tem uma grande probabilidade de alcançar o objetivo pretendido. O facto de todos elementos partilharem entre si a participação ativa, vai conceber um ambiente de

aprendizagem facilitador da participação ativa dos alunos. Desta forma todos os elementos vão ter um papel ativo no processo de ensino-aprendizagem.

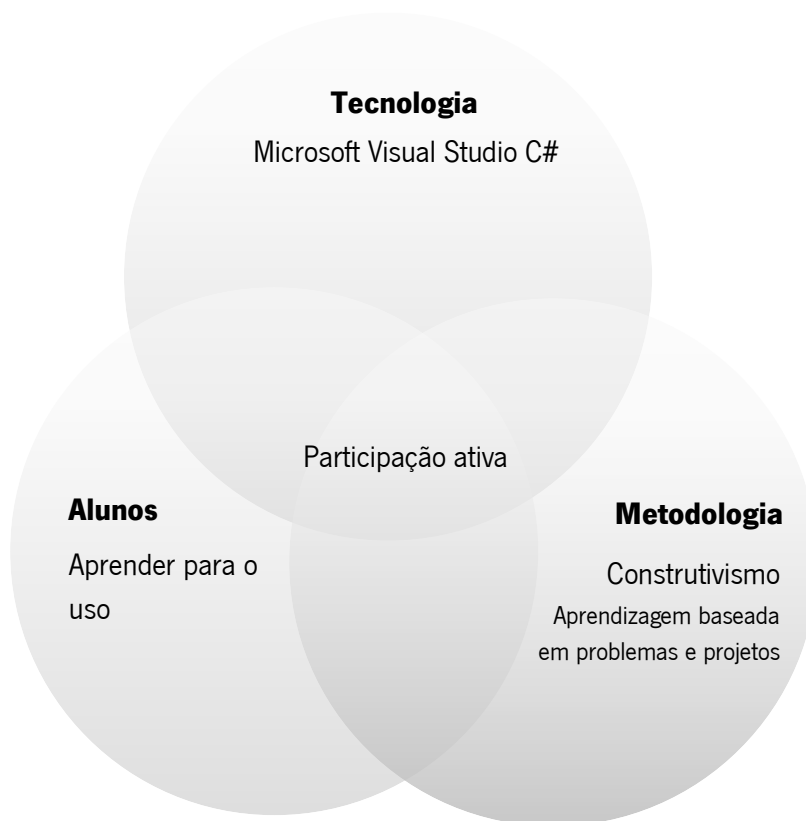


Figura 4 - Conceção de um ambiente de aprendizagem que promova a participação ativa

Um outro exemplo, se os alunos receberem um feedback imediato, e estiverem a resolver problemas próximos da realidade tem tendência a estar mais motivados para aprender. Partindo deste princípio, procurou-se uma tecnologia que fornecesse um feedback imediato e lhes permitisse resolver esse tipo de problemas, assim valida-se novamente a aplicação Microsoft Visual C#, pois esta retorna um feedback imediato ao aluno, apresentando a solução ou demonstrando onde está o erro e permite a resolução de problemas reais ou aproximados à realidade. Na escolha das metodologias teve-se o mesmo cuidado, trabalhando-se com metodologias baseadas em ideais construtivistas, para as quais o feedback imediato das aprendizagens é fundamental, assim como a simulação de contextos próximos do real.

Este cuidado na escolha das tecnologias e metodologias, que vão de encontro às características, dificuldades e necessidades de aprendizagem dos alunos, esteve sempre presente ao

longo do Projeto de Intervenção Pedagógica. Todo este processo permitiu que as metodologias e tecnologias utilizadas fossem sempre adequadas a cada momento específico de aprendizagem.

Se existir intersecção, as tecnologias vão de encontro e acrescentam valor à metodologia e, ao mesmo tempo, respeitam as necessidades e preferências dos alunos. O mesmo se sucede com as metodologias que vão permitir o uso e rentabilização da tecnologia em prol do sucesso das aprendizagens dos alunos. Acredito que, quanto maior for a intersecção entre os elementos, maior é a probabilidade de o ambiente de aprendizagem cumprir o seu propósito.

2.2 O projeto desenvolvido pelos alunos ao longo da intervenção

Partindo do processo anteriormente descrito, e tendo em atenção a complexidade dos conteúdos programáticos a serem lecionados ao longo deste projeto, a tecnologia utilizada e a não menos importante complexidade geral da programação, decidi, apoiado na metodologia de aprendizagem baseada em projetos, conceber um projeto para os alunos desenvolverem ao longo da intervenção. Numa primeira fase pensei no tipo de aplicação que gostaria que os alunos desenvolvessem, sendo os alunos de um curso profissional, e a disciplina Programação e Sistemas de Informação, surgiu-me a ideia de conceber um projeto baseado no desenvolvimento de uma aplicação de gestão. Para além das razões já enumeradas, tive em atenção no seu desenho outras mais específicas, tais como:

- proximidade com a realidade, seguindo os pressupostos construtivistas, concebi um projeto em tudo semelhante, àqueles que os alunos um dia terão de desenvolver nas empresas;
- versatilidade, de modo a permitir que os alunos se envolvessem, e criativamente seguissem o seu caminho dentro do mesmo;
- poder de escolha, quer de temas quer de funcionalidades, valorizando-se assim os interesses e motivações dos alunos.

Numa segunda fase implementei a ideia para ter a certeza que os alunos seriam capazes de a concretizar.

Este projeto é um dos elementos mais importantes desta intervenção na medida em que esteve presente ao longo de todo o processo e serviu como o elemento principal da sua avaliação.

No plano de intervenção tinha delineado que os projetos seriam desenvolvidos por grupos de dois alunos, mas depois de refletir sobre o assunto, optei pela resolução individual. Isto sucedeu por

uma serie de razões que serão apresentadas de seguida. Em primeiro lugar, os alunos eram poucos (12) e existiam estações de trabalho (computadores) para todos, logo não existia a limitação do número de postos de trabalho. Em segundo lugar, porque a tendência na programação, quando dois alunos trabalham em grupo é um tomar uma postura ativa e outro uma postura passiva na resolução do projeto, e eu quis essencialmente implementar uma abordagem onde todos adotassem uma postura ativa no seu processo de aprendizagem. Num grupo existe sempre um aluno que dá o primeiro passo, geralmente é sempre o aluno com mais conhecimentos, e o problema surge quando “os outros acompanham e acreditam veemente que sozinhos conseguem resolver o problema desde o início” (Brito & Sá-Soares, 2010, p. 224). A solução poderia passar pelo trabalho cooperativo, mas assim não estariam a concretizar todas as aprendizagens ao nível da programação, cada um só concretizaria a sua parte. Em terceiro lugar, como os alunos eram poucos, era mais fácil orienta-los e detetar as suas dificuldades, pois em grupo facilmente conseguiriam oculta-las. Em quarto lugar, seria mais fácil medir a aprendizagem e o progresso individual dos alunos quando estes trabalham individualmente. Em quinto lugar, porque ao longo deste projeto de intervenção trabalhei mais o lado cognitivo do construtivismo e menos o social.

O suporte teórico da conceção deste projeto de gestão esteve no paradigma construtivista e nas metodologias de ensino da informática na Dinamarca, mais propriamente a *stepwise improvement* e *worked examples*. Tendo por base estas metodologias decidi dividir o projeto em várias tarefas, que inicialmente tinham um elevado grau de dependência do professor, a qual se foi desvanecendo ao longo do tempo, com o objetivo de ir promovendo a autonomia dos alunos.

Uma outra razão que me levou a subdividir o projeto foi o facto de, durante o período de observação, antes da intervenção, ter verificado que os alunos, quando os professores lhes apresentavam um projeto de grandes dimensões, não serem bem-sucedidos principalmente devido à dificuldade de criar pontos de referência e à complexidade do todo. Com esta estratégia procurei que os alunos conseguissem autorregular-se, situando-se no ponto onde se encontravam relativamente à resolução do projeto, tendo por referência as tarefas.

O projeto de gestão foi então dividido em 4 tarefas, como se pode visualizar na seguinte imagem.



Figura 5 - Divisão em tarefas do projeto a ser desenvolvido pelos alunos

Como já foi referido anteriormente a disciplina tinha uma carga horária de dois blocos de 90, em três dias diferentes. Para cada tarefa, como estratégia, nos primeiros 90 minutos (1ª fase) os alunos construíram e reestruturaram os seus conhecimentos segundo diversas atividades e metodologias construtivistas. Nos segundos 90 minutos (2ª fase), tiveram a oportunidade de implementar, testar e alterar os conhecimentos adquiridos na fase anterior, através da realização do projeto. Deste modo, durante os primeiros 90 minutos, sendo os alunos sempre a parte ativa do processo, realizaram-se as aprendizagens dos conceitos e foram resolvidos alguns desafios (problemas) com a orientação do professor, colocando em prática os conhecimentos adquiridos, relacionando-os com os conhecimentos prévios resultantes da sua interação com a realidade.

Nos segundos 90 minutos os alunos aplicaram autonomamente os conhecimentos adquiridos nos 90 minutos anteriores (construtivismo) a uma nova situação, concebendo, para tal, o projeto tarefa a tarefa. A cada nova tarefa os alunos iam expandindo e reestruturando o seu produto final.

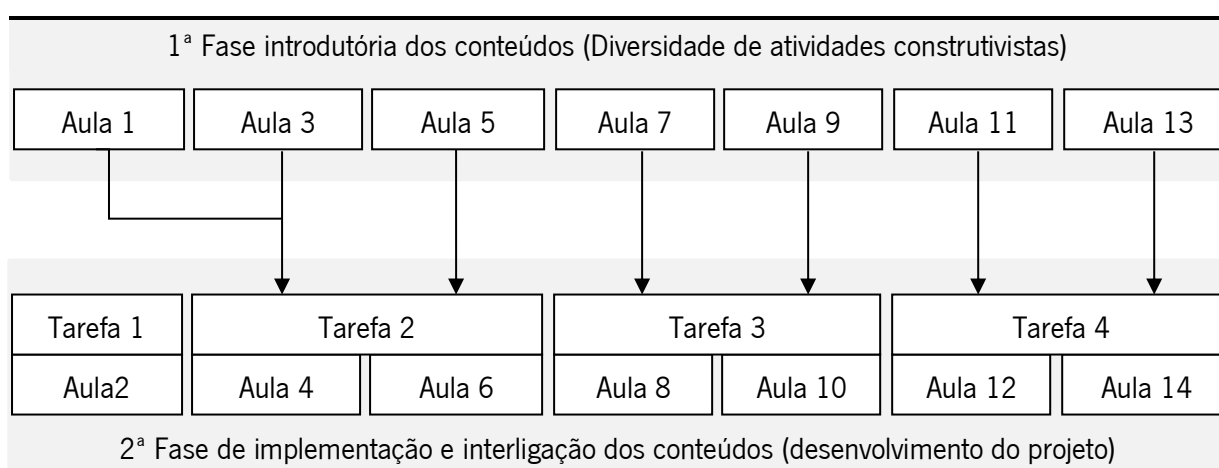
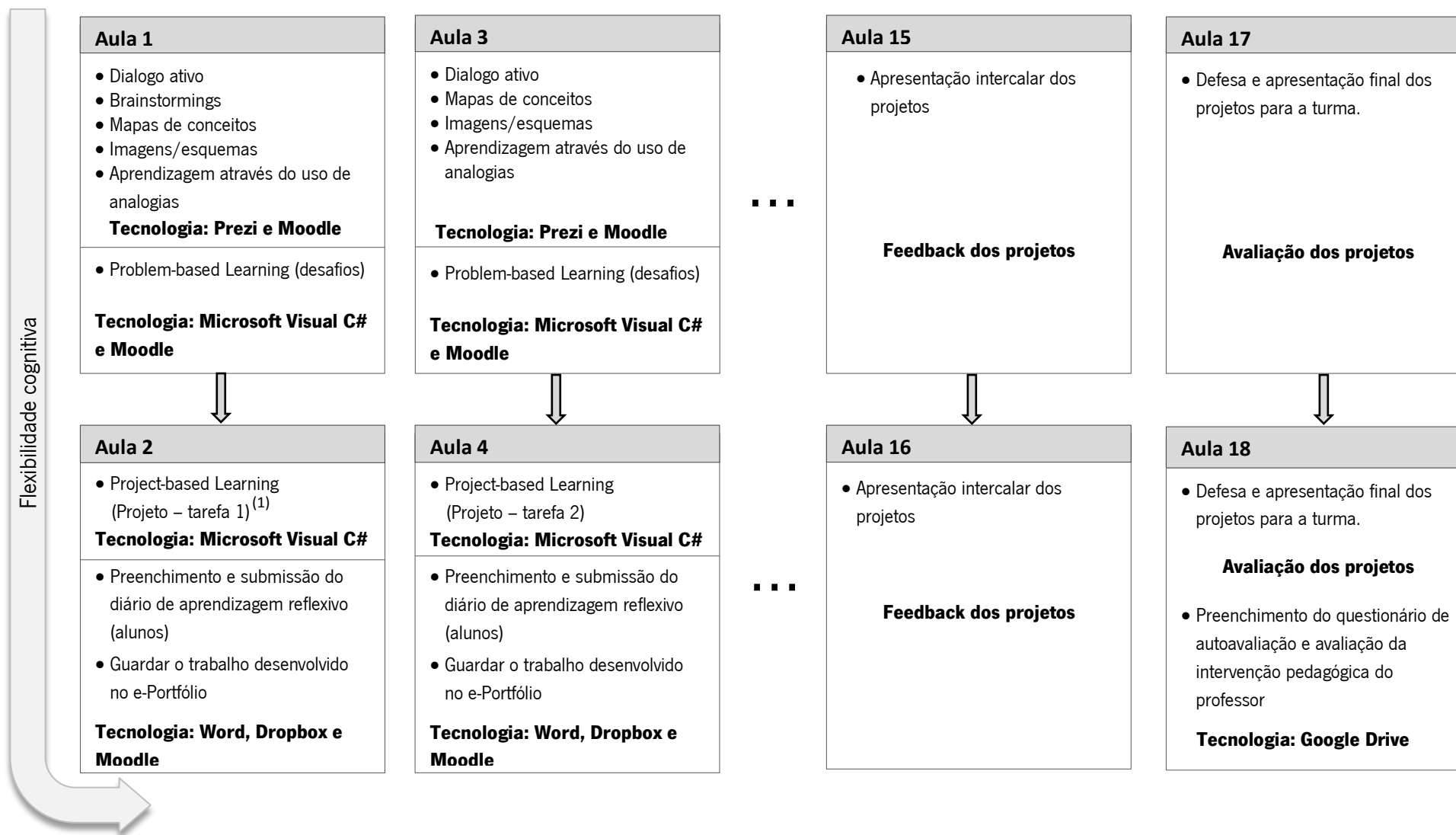


Figura 6 - Esquema exemplificativo da relação entre as aulas

Ao voltarem a aplicar as aprendizagens concretizadas durante a primeira fase, a um novo contexto (segunda fase) os alunos desenvolveram a sua flexibilidade cognitiva.

Para ajudar a compreender melhor este projeto de intervenção e a forma como a abordagem construtivista foi aplicada, elaborei o seguinte esquema, onde constam as metodologias, estratégias e tecnologias utilizadas ao longo das aulas.

Síntese da intervenção



¹ Segundo as metodologias do ensino da informática na Dinamarca: *stepwise improvement* e *worked example*

2.3 Tecnologia educativa

Tendo como objetivo principal a aprendizagem de conteúdos relacionados com Programação Orientada a Objetos avançada, promovendo-se para esta concretização a motivação e autonomia dos alunos, foi utilizada ao longo desta intervenção a tecnologia educativa que seguidamente se descreve.

A sala de aula estava bem equipada ao nível dos recursos educativos, dispunha de 14 computadores com acesso à internet, permitindo que cada aluno tivesse disponível um computador ao longo de todo o projeto. Um projetor fixo direcionado para um quadro interativo, cuja utilização implicava o uso do *software ActivInspire*, sempre que ao longo de uma atividade os alunos precisassem de escrever sobre a projeção.

2.3.1 Prezi

O *Prezi* é uma ferramenta da *web 2.0* que permite a criação de apresentações não lineares dinâmicas e criativas de alto impacto através do conceito de apresentações de *zoom* numa única estrutura.

Esta tecnologia sugere um conjunto de *templates*, de entre as opções disponibilizadas escolhi o “*the office*” (figura 7) por se enquadrar no curso e na matéria a lecionar, pois apresenta o quadro branco para a matéria teórica, os post-its para informações adicionais, os computadores para as partes práticas, exercícios e desafios e os dossiês para mostrarem e agruparem informações relacionadas com diretorias e ficheiros.

Com base nesta tecnologia foi criada uma apresentação que foi utilizada ao longo de toda a intervenção.

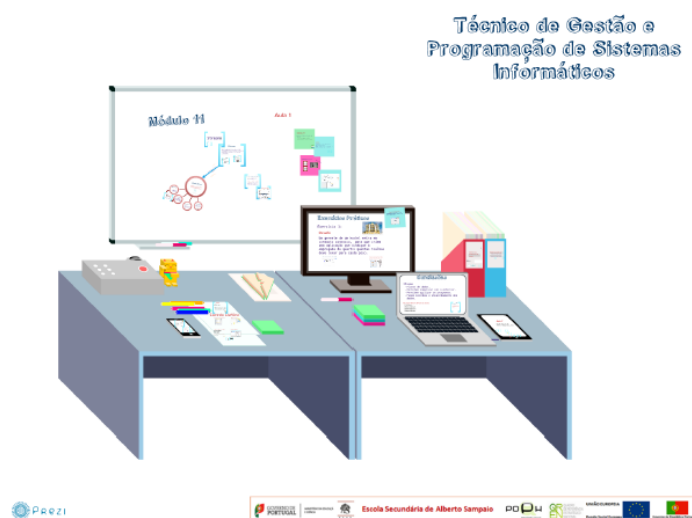


Figura 7 - Template do prezi escolhido para a intervenção

2.3.2 Plataforma moodle

A plataforma moodle foi utilizada ao longo desta intervenção para disponibilizar aos alunos todos os materiais utilizados durante as aulas. Os materiais foram organizados por aula (figura 8), onde também constava o sumário. A utilização desta plataforma foi extremamente importante na medida em que por um lado, os alunos faltam muito, e assim tinham acesso a todos os materiais utilizados a partir de casa, e por outro lado, como eles são extremamente desorganizados, nunca trazem mochila, nem cadernos, foi uma forma de terem sempre tudo organizado. A plataforma também serviu para no final das aulas os alunos submeterem o diário de aprendizagem.

A minha aposta na utilização desta plataforma também surgiu do facto de, durante período de observação e análise documental, ter constatado que todos os alunos tinham computador e acesso à internet em casa. Isto permitiu expandir a sala de aula até casa dos alunos, permitindo-lhes consultar os materiais e assim continuar a desenvolver o seu projeto em casa.

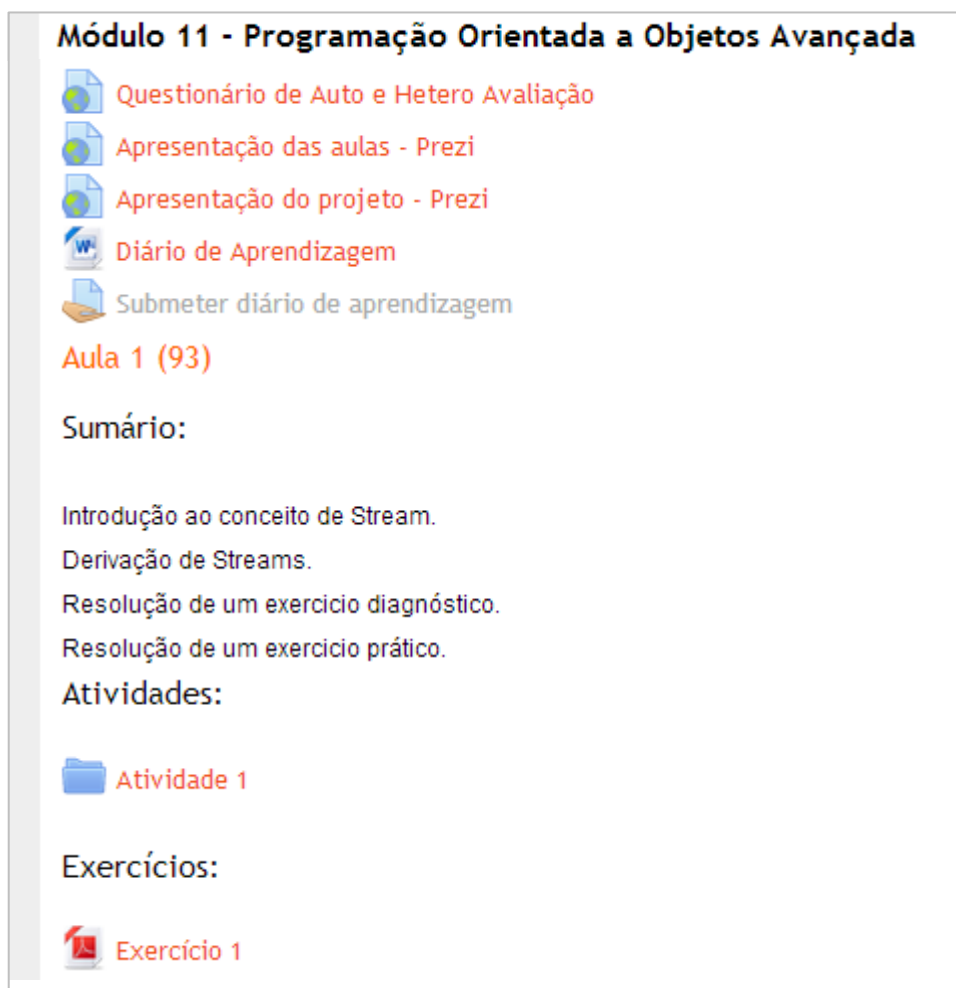


Figura 8 - Organização da plataforma moodle

2.3.3 Microsoft Visual C# 2010

Como a programação se aprende essencialmente através da prática, é fundamental escolher a tecnologia adequada, recorrendo à técnica da intersecção dos elementos educativos anteriormente descrita escolheu-se a plataforma de desenvolvimento Microsoft Visual C# 2010. Segundo Kordaki (2010), os alunos fomentam as suas competências em programação quando assistidos por tarefas de instrução e tecnologias educacionais adequadas.

Esta plataforma adequa-se na perfeição a esta intervenção na medida em que:

- Possibilita o desenvolvimento de aplicações com base no paradigma da Programação Orientada a Objetos na linguagem C#.
- Permite a criação de aplicações interativas em Windows forms, através da manipulação de componentes de forma visual por *drag-and-drop*.
- É uma ferramenta atual utilizada no mercado de trabalho.
- Incentiva a autonomia e a criatividade.
- Permite um processo de ensino aprendizagem centrado no aluno.
- Obedece aos pressupostos construtivistas, fornece um feedback imediato aos alunos e permite a valorização do erro.

A tecnologia fornece dois tipos de feedback (figura 9), o primeiro, através do processo de execução do programa que apresenta o resultado do mesmo, permite ao aluno comprovar se as suas ideias e soluções estão corretas e cumprem relativamente ao planeado. Caso a solução não seja a esperada, o aluno pode refletir e regressar ao código para reformular o seu programa. Este feedback é fiel e imediato e permite ao aluno rever e reformular ciclicamente a sua solução até obter a resposta esperada. Através deste processo de construção e desconstrução, baseado na resposta do computador, os alunos vão construindo novos conhecimentos e desenvolvendo a sua flexibilidade cognitiva.

O segundo feedback ocorre quando o programa informático desenvolvido pelos alunos apresenta erros de programação, nesta situação a aplicação avisa da existência do erro, indicando a sua localização. Os alunos começam a ver os erros não como algo sancionatório, mas sim como algo que os ajuda a aprender e a reformular as suas soluções. Ao examinar as causas dos erros, os alunos são incentivados a refletir, e dentro desta reflexão reestruturam as representações internas do seu conhecimento.

Este feedback tecnológico ajuda os alunos a tomar consciência sobre o que sabem e não sabem e a experimentar diferentes táticas e estratégias com a intenção de descobrirem a melhor forma de ultrapassar os obstáculos.

Esta tecnologia permitiu a eficaz implementação de metodologias construtivistas, como a aprendizagem baseada em problemas e a aprendizagem baseada em projetos, pois permitiu que de uma forma ativa os alunos desenvolvessem as suas soluções.

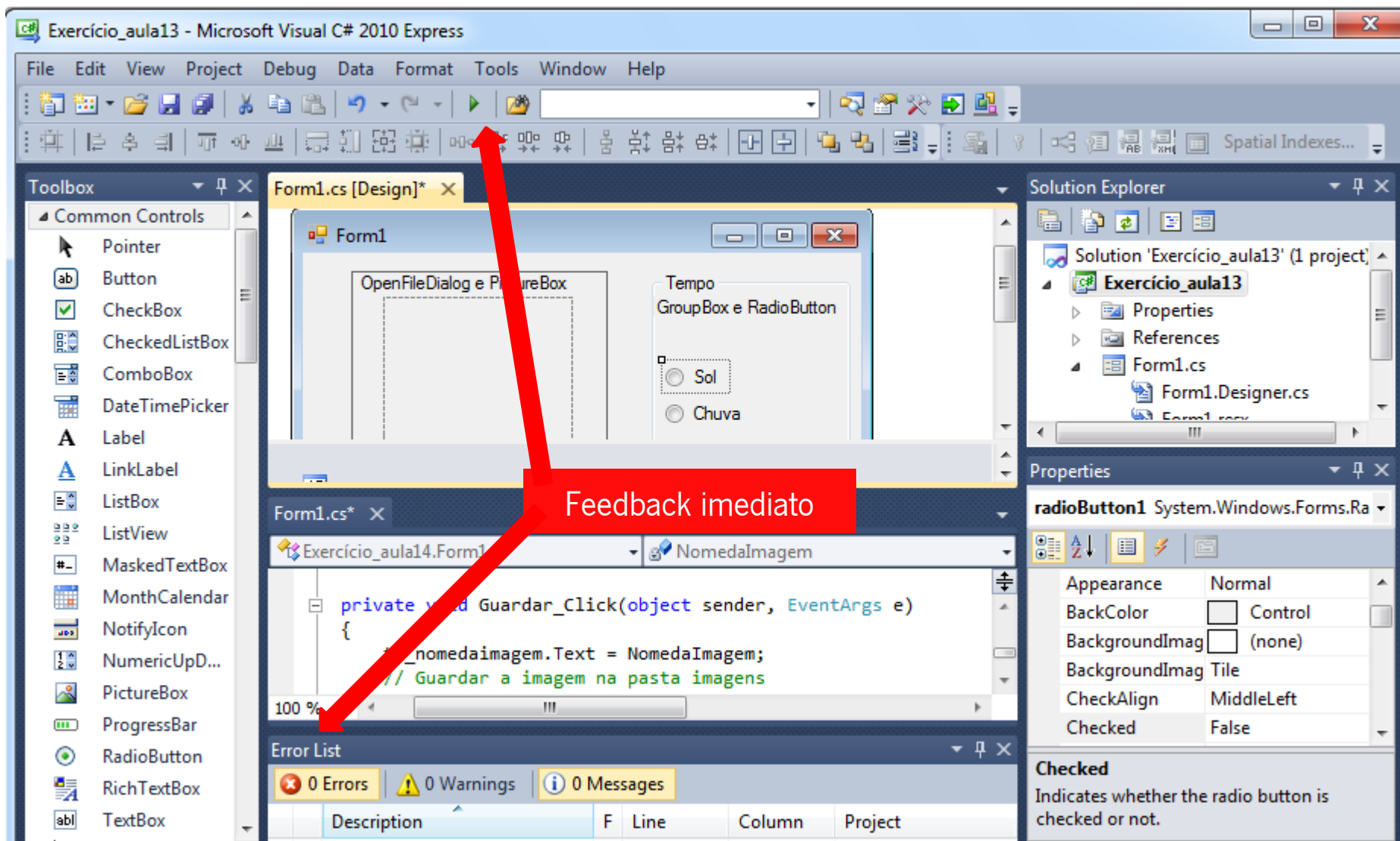


Figura 9 - Feedback da tecnologia Microsoft Visual C# 2010

2.3.4 Dropbox

Tendo por base a minha experiência como professor em cursos profissionais, 5 anos a lecionar em escolas secundárias, a observação do comportamento dos alunos, e o facto de a maioria das atividades projetadas para esta intervenção se prolongarem por várias aulas, sendo necessário que os alunos tenham presente o trabalho das aulas anteriores, antevi que os alunos estariam constantemente a perder os trabalhos, ou a esquecerem-se deles em casa, comprometendo assim o seu trabalho durante as aulas. No sentido de ultrapassar esta situação, após alguma investigação, encontrei o que achei ser a melhor solução: a *Dropbox*.

A *Dropbox* é um serviço de armazenamento de arquivos, baseado no sistema da “nuvem”, que permite aos alunos ter acesso aos seus ficheiros em qualquer lugar, desde que tenham acesso à internet.

Cada aluno criou e geriu uma conta na Dropbox, a qual serviu de e-portfólio e lhes permitiu guardar os materiais produzidos ao longo das aulas e os seus diários reflexivos sobre a aprendizagem.

2.4 Atividades e estratégias realizadas no âmbito do projeto

Calendarização e enumeração das principais atividades e estratégias realizadas no âmbito do projeto				
Aula	Duração	Data	Conteúdos	Atividades/estratégias
1	90'	23-01-2013	<i>Streams</i>	<ul style="list-style-type: none"> – Brainstorming sobre o conceito de stream e streaming. – Atividade colaborativa de associação e reconstrução do conhecimento. Aprendizagem por analogia "Comboio versus <i>Stream</i> de <i>input</i> e <i>output</i>". – Atividade "Mãos na massa" - Exploração, teste e reflexão sobre o processo de carregamento de ficheiros para memória RAM (<i>FileStream</i> e <i>MemoryStream</i>) recorrendo à Microsoft Visual Studio C# 2010 e ao gestor de tarefas. – Atividade colaborativa de consolidação de conhecimentos relativamente à classe <i>BufferedStream</i> – Aprendizagem por analogia com o ciclo da água. – Resolução de um exercício diagnóstico. – Resolução de um problema (Desafio - Hotel) – <i>problem-based learning</i>.
2	90'	23-01-2013		<ul style="list-style-type: none"> – Projeto - Resolução da tarefa 1. – Criação de uma conta na Dropbox. – Preenchimento do diário de aprendizagem.
3	90'	24-01-2013	Ficheiros de texto	<ul style="list-style-type: none"> – Análise colaborativa de mapas mentais para recapitulação/ligação de conceitos. – Resolução do problema "Nomes e apelidos" – <i>problem-based learning</i>.
4	90'	24-01-2013		<ul style="list-style-type: none"> – Projeto - Resolução da tarefa 2. – Preenchimento do diário de aprendizagem.
5	90'	29-01-2013	Ficheiros de texto	<ul style="list-style-type: none"> – Resolução do problema "Ficheiros de texto" – <i>problem-based learning</i>.

Calendarização e enumeração das principais atividades e estratégias realizadas no âmbito do projeto				
Aula	Duração	Data	Conteúdos	Atividades
6	90'	29-01-2013		<ul style="list-style-type: none"> – Projeto - Resolução da tarefa 2 (conclusão). – Preenchimento do diário de aprendizagem.
7	90'	30-01-2013	Ficheiros binários Dicionários	<ul style="list-style-type: none"> – Atividade colaborativa - quadro interativo (ficheiros binários). – Atividade colaborativa e cooperativa (Desafio Call Center) – problema-based learning.
8	90'	30-01-2013		<ul style="list-style-type: none"> – Projeto - Resolução da tarefa 3. – Preenchimento do diário de aprendizagem.
9	90'	31-01-2013	ListView	<ul style="list-style-type: none"> – Atividade Desafio Call Center upgrade (ListView) - problem -based learning.
10	90'	31-01-2013		<ul style="list-style-type: none"> – Projeto - Resolução da tarefa 3 (conclusão).
11	90'	05-02-2013	Serialização	<ul style="list-style-type: none"> – Atividade construtivista de construção de conhecimento por analogia (sapo insufável versus serialização). – Atividade Desafio Call Center upgrade (Serialização) - <i>problem -based learning</i>.
12	90'	05-02-2013		<ul style="list-style-type: none"> – Projeto - Resolução da tarefa 4.
13	90'	06-02-2013	Controlos	<ul style="list-style-type: none"> – Atividade de diálogo e apresentação de vários controlos.
14	90'	06-02-2013		<ul style="list-style-type: none"> – Projeto - Resolução da tarefa 4 (conclusão).
15	90'	07-02-2013		<ul style="list-style-type: none"> – Apresentação dos projetos (<i>feedback</i>).

Calendarização e enumeração das principais atividades e estratégias realizadas no âmbito do projeto				
Aula	Tempo	Data	Conteúdos	Atividades
16	90'	07-02-2013		– Apresentação dos projetos (<i>feedback</i>)
17	90'	14-02-2013		– Apresentação e defesa dos projetos (final)
18	90'	14-02-2013		– Apresentação e defesa dos projetos (final)

Tabela 1 – Calendarização e distribuição das atividades

Ao longo desta secção serão descritas de forma sintética as várias atividades concretizadas pelos alunos durante este projeto de intervenção pedagógica e algumas conclusões que foram surgindo.

2.4.1 Aulas 1 e 2

Nesta primeira aula, que iniciou a minha prática letiva, tendo por referência a planificação do módulo 11 (anexo 1), comecei por trabalhar com os alunos o conceito de *stream*. Para Marques, Pedroso e Figueira (2011), uma *stream* representa um fluxo de informação que se desloca de/para algum lado. Uma *stream* de entrada representa uma fonte de informação e uma *stream* de saída representa algo que envia informação para um determinado destino.

Sendo este um conceito complexo de entender, essencialmente ao nível do seu funcionamento na prática, para uma melhor compreensão, tendo sempre presente a promoção da autonomia e motivação, recorreu-se a uma série de atividades, estratégias e metodologias, as quais serão apresentadas de seguida.

Brainstorming sobre o conceito de stream e streaming

Com a intenção de aproveitar a opinião dos alunos para a clarificação do conceito de stream, como primeira atividade para a construção deste conhecimento escolhi um brainstorming. Escolhi esta atividade, por me parecer que de uma forma construtivista permite que os alunos colaborativamente construam o seu próprio conhecimento a partir dos seus conhecimentos prévios, da sua interação com a realidade e com os colegas.

Dei início ao brainstorming com a apresentação de uma imagem (figura 10), recorrendo à apresentação eletrónica em *Prezi*. De seguida solicitei aos alunos que proferissem a sua opinião sobre o que entendiam por *stream* e *streaming*.



Figura10 - Momento do prezi que fomentou o brainstorming

Com o decorrer do brainstorming os alunos foram chegando à conclusão pretendida, ou seja, que sempre que estão a ver um vídeo *online*, está a decorrer o processo de *streaming*, e que este lhes permite enviar e receber dados de/para o computador.

Ao longo da atividade os alunos estiveram muito participativos, às vezes até exageradamente criando algum ruído na sala, o que é normal num ambiente construtivista onde prevalece o diálogo/discussão. Macedo afirma, relativamente a este aspeto em concreto que um ambiente construtivista ao contrário de um tradicional, que "pede o silêncio e a contemplação do docente... pede o ruído e a manipulação nem sempre jeitosa daqueles que, aceitando uma pergunta, não estão satisfeitos com o nível de suas respostas" (1993, p. 30).

É importante referir, que foi necessário durante o processo moderar a participação dos alunos, pois a ala direita da turma estava mais participativa, sendo necessário fomentar e dar também oportunidades de participação à ala esquerda. Tive este cuidado porque o que acontece com bastante frequência é que um ou mais alunos da turma, normalmente os melhores alunos, dominam o diálogo impedindo os restantes de participar.

Concluí através da observação desta atividade que o brainstorming é uma atividade de partilha, que potencia a participação, troca de ideias e perspetivas que permitem aos alunos acomodar novas informações e reformular as suas compreensões sobre o tema. Esta chuva de ideias surge porque, segundo Mullen, Johnson e Malas (1991), num grupo, a ideia de um dos elementos pode despertar outro, fazendo com que todos ampliem a capacidade normal de pensar sobre um determinado assunto ou questão, no fundo opinião gera opinião.

De seguida, mantendo este ambiente de diálogo e postura ativa dos alunos no processo de ensino-aprendizagem, procurando reforçar e consolidar o conceito de streaming, foi apresentado um esquema aos alunos (figura 11). Pedi aos alunos que analisassem e explicassem o que viam no esquema, permitindo-lhes assim fazer a ligação entre conceito de stream e o conceito de canal ou fluxo de informação entre a aplicação e o mundo exterior.



Figura11 - Esquema demonstrativo do funcionamento das streams

Atividade colaborativa de associação e reconstrução do conhecimento. Aprendizagem por analogia "Comboio versus Stream de input e output".

Para facilitar o processo de compreensão, elaboração e construção do conceito de *stream* de *input* e *output*, visto este ser um conteúdo considerado complexo, decidi para ser mais fácil a sua compreensão, criar uma analogia entre os conceitos de stream de input e output e algo que os alunos já conhecessem. Assim surgiu a ideia da analogia com um comboio de transporte de mercadorias

A analogia foi apresentada recorrendo à apresentação prezi, como se pode verificar na seguinte imagem.



Figura12 - Analogia como facilitadora da aprendizagem de Streams de input e output

Após a apresentação da analogia solicitei aos alunos que identificassem as semelhanças entre os dois processos. Muitas vezes, no sentido de os orientar para o propósito da aprendizagem através do uso desta analogia, tive de colocar questões cognitivamente direcionadoras, como por exemplo, “nesta imagem do comboio quem acham que é a aplicação”, “quem é a stream, e qual é a sua função”.

Verifiquei através das respostas/intervenções dos alunos que estes associaram o conhecimento que já possuíam sobre comboios, os quais transportam mercadorias de uma origem para um destino ou vice-versa, com o conceito de *stream*, responsável pelo transporte de informação de/para uma aplicação informática. Também associaram o facto da pessoa responsável pelo transporte das mercadorias no comboio ser o maquinista e não eles, ao conceito de virtualização do fluxo de informação nas streams, onde estas são responsáveis por todos os processos inerentes à comunicação, libertando assim o utilizador de qualquer responsabilidade ou trabalho.

Outro conceito que os alunos relacionaram com sucesso, foi a grande quantidade de mercadoria transportada pelo comboio, e a possibilidade de manuseamento de grandes quantidades de informação pelas *streams*.

A aprendizagem por analogia permitiu-lhes relacionar os novos conceitos de grau de complexidade elevado com conhecimentos que eles já possuíam, comparando assim dois domínios diferentes, um conhecido e outro desconhecido.

Durante esta fase, enquanto professor, procurei ter um papel orientador e facilitador fornecendo apenas sugestões que visaram estimular o seu pensamento.

Atividade "Mãos na massa" - Exploração, teste e reflexão sobre o processo de carregamento de ficheiros para memória RAM (FileStream e MemoryStream) recorrendo à Microsoft Studio C# 2010 e ao gestor de tarefas.

Esta atividade surgiu para que os alunos construíssem os seus conhecimentos sobre as classes derivadas da classe abstrata stream: *FileStream*, *MemoryStream* e *bufferedStream*.

Para tal, inicialmente recorrendo à apresentação *Prezi*, apresentei as classes *FileStream* e *memoryStream*, como podemos verificar na seguinte imagem.

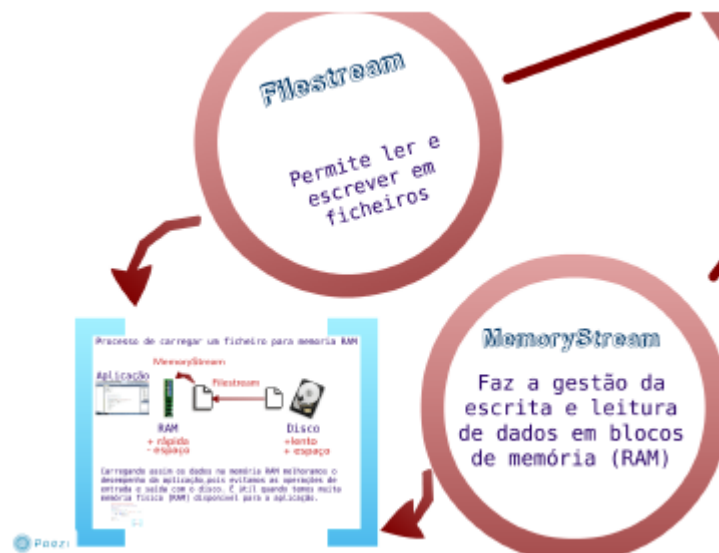


Figura13 - Introdução dos conceitos de filestream e memorystream

Para melhor compreensão do funcionamento destas classes, criei um esquema exemplificativo para que os alunos visualisassem como elas funcionam na prática. Com este esquema procurei que eles estabelecessem relações com aprendizagens já concretizadas, refletindo e explicando o processo descrito no esquema e, assim entendessem que carregando em memória RAM os dados, o desempenho da aplicação melhora, evitando sucessivos acessos ao disco.

O esquema foi apresentado no *Prezi* como podemos verificar na seguinte imagem.

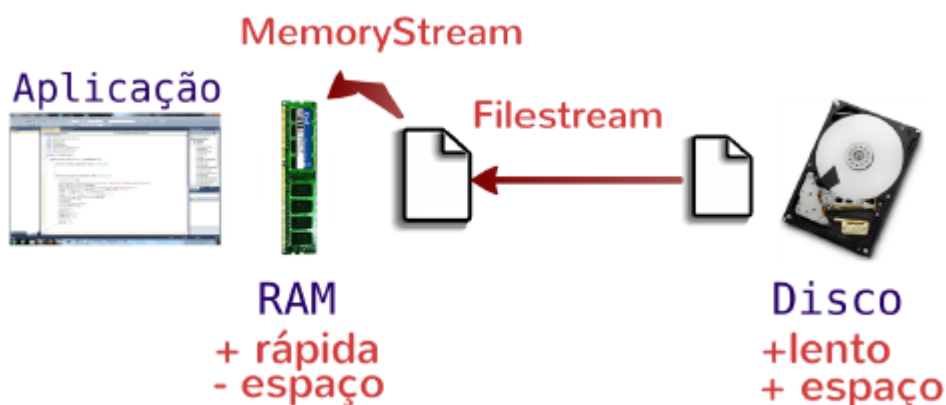


Figura14 - Processo de carregar um ficheiro para memória RAM

Após, os alunos terem compreendido a teoria das classes e o esquema de funcionamento das mesmas, decidi avançar e apresentar aos alunos o código representativo deste processo (Figura 15). De uma forma colaborativa os alunos analisaram e comentaram o seguinte código:

Como funcionará isto na prática ???

Exemplo prático de leitura de um ficheiro para memória RAM.

```
MemoryStream memstream = new MemoryStream();  
FileStream ficheiro = new FileStream(@"C:\Ficheiros\bibliografia.docx", FileMode.Open)  
byte [] arraybytes = new byte[ficheiro.Length];  
ficheiro.Read(arraybytes, 0, (int)ficheiro.Length);  
memstream.Write(arraybytes, 0, (int)ficheiro.Length);  
ficheiro.Close();  
memstream.Close();
```

Figura15 - Código de leitura de um ficheiro para memória RAM

Depois de analisado e compreendido o código, solicitei aos alunos que realizassem uma atividade onde vissem a aplicabilidade deste código. Para tal, tiveram de aceder à plataforma moodle e descarregar a aplicação criada propositadamente para o efeito. A aplicação efetuava a leitura de dois ficheiros para a memória RAM, um pequeno com alguns Kbs e outro maior com 2 Mbs.

Mãos na massa ...



Abram o moodle e vamos testar a
aplicação da atividade 1

Figura16 - Momento do prezi em que é solicitado aos alunos que realizem a atividade

Adotando a postura ativa no processo de ensino-aprendizagem os alunos descarregaram a aplicação da plataforma moodle e executaram-na na Plataforma Visual Studio C#. Esta aplicação teve que ser executada duas vezes, uma por ficheiro.

Recorrendo ao gestor de tarefas, após cada execução da aplicação, analisaram a memória RAM ocupada por cada um dos ficheiros, fazendo a comparação com a memória ocupada pela aplicação antes de serem carregados os ficheiros. Desta forma, os alunos tiveram a oportunidade de confirmar que realmente os ficheiros são carregados em memória RAM.

Atividade colaborativa de consolidação de conhecimentos relativamente à classe `BufferedStream` – Aprendizagem por analogia com o ciclo da água.

Concluída a atividade “Mãos na massa”, introduziu-se um conteúdo novo, a classe `bufferedStream`, para a qual foi apresentado um esquema representativo do seu funcionamento, tal como tinha sido apresentado para a classe `fileStream` e `memoryStream`. Mais uma vez foi solicitado aos alunos que explicassem o fluxo de informação do seguinte esquema.

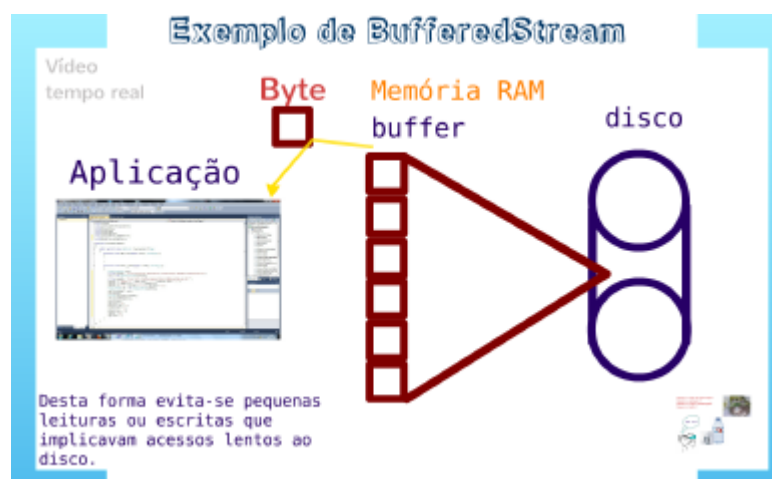


Figura17 - Exemplo de funcionamento da classe `bufferedStream`

Como já foi referido na aprendizagem do conceito de stream, para que os alunos construam o seu conhecimento (construtivismo), tendo por base a realidade já conhecida, foi apresentada uma representação do real muito análoga, ao esquema anterior. Para tal, pensei em algo que os alunos já conhecessem e lhes permitisse compreender melhor este conteúdo, e assim, cheguei à analogia entre o funcionamento da classe `bufferedStream` e um pequeno circuito da água entre a fonte e o utilizador final.

Esta analogia permitiu aos alunos construírem o seu conhecimento sobre o conteúdo `BufferedStream` associando a fonte de água a uma fonte de informação (ex. disco); o garrafão a um `buffer` (responsável por guardar um conjunto de dados obtidos da fonte); o copo de água a um bloco de informação; e uma pessoa à aplicação que vai necessitar de consumir dados. Esta analogia também possibilitou que os alunos, se consciencializassem sobre as vantagens da utilização do `buffer`, pois a

utilização de um garrafão de água (*buffer*), evita que cada vez que se tenha sede, seja necessário ir à fonte buscar água. O mesmo sucede com o *buffer*, que permite que os dados necessários à aplicação estejam mais perto da mesma, evitando assim constantes acessos ao disco, os quais tornam as aplicações mais lentas.

Para verificar se os alunos compreenderam a analogia foi pedido aos mesmos que respondessem às seguintes questões:



Figura18 - Analogia com o ciclo da água

Resolução de um exercício diagnóstico

Transitando para uma parte mais prática da aula, foi solicitado aos alunos que resolvessem um exercício prático de diagnóstico, a fim de ser possível aferir os seus conhecimentos relativamente à plataforma Microsoft Visual Studio C# 2010, às bases algorítmicas, à programação em C# e ao paradigma da Programação Orientada a Objetos.

Pedi então aos alunos que criassem uma simples aplicação Windows Forms que percorresse um *array* com 4 elementos do tipo inteiro e devolvesse o maior.

Ao longo da sua resolução verifiquei que os alunos tiveram muitas dificuldades, apenas um aluno conseguiu concluí-lo, tendo necessitado da minha ajuda para o conseguir. Pedi a esse aluno que fosse ao quadro explicar aos restantes colegas a sua solução, por reconhecer que os alunos aprendem uns com os outros.

Analisando e refletindo os resultados do exercício diagnóstico a verdade é que nenhum dos alunos conseguiu resolver o exercício sozinho, e como o exercício era bastante simples, isto preocupou-

me seriamente, pois tinha de lecionar o último módulo de programação, ou seja a “última maratona”, quando estes alunos demonstravam que ainda nem sabiam caminhar.

Estava assim perante um desfasamento muito grande entre os conhecimentos dos alunos e o nível no qual deveriam estar neste momento. Os alunos não tinham praticamente bases nenhuma no nível da programação, apresentavam sérias dificuldades ao nível dos ciclos e do raciocínio abstrato, resultantes das dificuldades ao nível da algoritmia.

É importante ter em consideração, neste ponto, que a apropriação, ou não, dos conceitos iniciais de programação tem uma relação direta com o desempenho do aluno no decorrer de todo o curso, já que módulos avançados dependem fortemente desses conceitos iniciais.

Com base nesta análise de dados tive de reformular a planificação das aulas de maneira a encontrar uma solução para ensinar e rever diversos conteúdos, para além dos que constavam na planificação do módulo, tentando assim suprir parte das dificuldades que os alunos apresentam relativamente à programação. Então reformulei as diversas atividades, exercícios, problemas e o projeto, no sentido de colmatar parte destas dificuldades.

Ao associar este facto, a todas as outras razões já descritas, que vão desde a falta de motivação aos problemas de comportamento, depreendi que estava perante um grande desafio.

Resolução de um problema (Desafio - Hotel) - problem-based learning

Concluído o exercício diagnóstico, deu-se início à resolução de um problema, para consolidação de aprendizagens e melhoramento da flexibilidade cognitiva dos alunos relativamente ao conceito de *stream*. Utilizando como estratégia a aplicação em prática dos conteúdos aprendidos ao longo da aula a uma nova situação, relacionando-os com outros conhecimentos de programação. Com este propósito decidi utilizar a metodologia de aprendizagem baseada em problemas.

Associei ao problema a palavra desafio, para fomentar a motivação e a competição nos alunos, aplicando assim a *gamification* em ambientes de aprendizagem. Segundo Deterding, Sicart, Nacke, O’Hara, & Dixon (2011), *gamification* consiste em ir buscar elementos dos jogos com o objetivo de envolver e motivar os alunos na concretização das atividades.

No desenho dos problemas tive o cuidado de planear a simulação de casos práticos, procurando assim promover o empreendedorismo e aproximação dos alunos ao contexto empresarial, preparando-os tanto para a formação em contexto de trabalho, que decorrerá no próximo ano letivo, como para um futuro profissional como técnicos informáticos.

Foi então apresentado aos alunos o seguinte problema, para o qual eles desenharam e implementaram uma solução em C# recorrendo à tecnologia Microsoft Visual Studio C#.

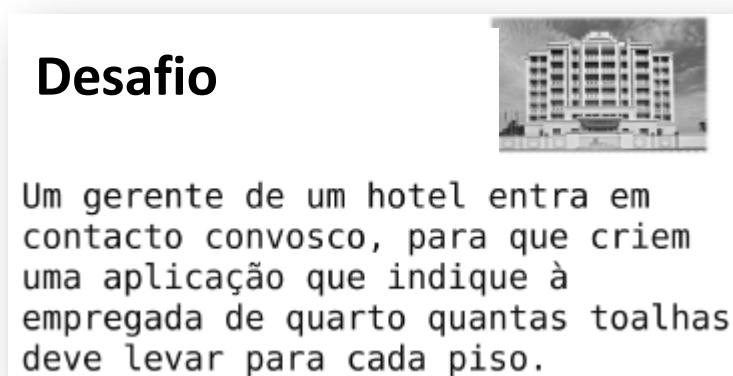


Figura19 - Problema do hotel

Todos os alunos demonstraram a capacidade de mobilizar os conhecimentos adquiridos ao longo da aula para esta nova situação, o que valida o sucesso das outras atividades na construção e compreensão do conceito de stream.

Apesar de todos os alunos terem concluído a atividade com sucesso, verifiquei que ainda eram pouco autónomos dependendo muito de mim para progredir nos exercícios e ao menor problema solicitavam a minha ajuda.

Projeto - Resolução da tarefa 1

Finalizada esta primeira aula de 90 minutos, deu-se início à segunda aula onde apresentei aos alunos o projeto que tiveram de desenvolver ao longo desta intervenção.

Como exemplo do projeto pretendido mostrei uma aplicação de gestão de uma equipa de futebol. Este tema foi apresentado unicamente como exemplo, e os alunos foram informados disso, pois para promover a sua autonomia e a motivação, apelando à motivação intrínseca, foi-lhes dada a liberdade de escolher o tema do seu projeto (ex.: gestão de um stand de automóveis, duma loja de informática, de jogos, etc.). Esta tomada de decisões permite uma maior envolvimento dos alunos no processo de ensino-aprendizagem. Desta forma os alunos desenvolveram os projetos tendo por base temas diferentes.

Para a apresentação do projeto criei um novo *Prezi*, no qual demonstrei aos alunos como seria subdividido o projeto, explicando-lhes que este seria subdividido em 4 tarefas e que cada nova tarefa acrescentaria novas funcionalidades e complexidade ao projeto. Esta divisão vai de encontro aos

princípios didáticos já descritos, utilizados no ensino da programação na Dinamarca, onde passo a passo se transfere a responsabilidade sobre as aprendizagens para os alunos e tarefa a tarefa vão ampliando e reestruturando a sua solução.

Foi dito aos alunos, com o intuito de os motivar, que cada tarefa resolvida é mais um passo para o sucesso, ou seja a aprovação no módulo. Como se pode ver na apresentação *Prezi* (figura 20), cada nova tarefa corresponde à subida de mais um degrau rumo ao sucesso académico e profissional.



Figura 20 - Prezi de demonstração do projeto

Ao longo da apresentação foram apresentados os conteúdos que seriam abordados para a concretização de cada tarefa.

Após ter apresentado o projeto à turma, informei os alunos que teriam de apresentar e defender a sua solução perante a turma, nas últimas aulas da intervenção.

Para cada tarefa criei um guião, onde constava o tempo recomendado para a sua resolução, tempo este encarado como uma sugestão que visava servir de orientação, os objetivos, uma visão geral da tarefa para contextualizar os alunos, ajuda à tarefa e sugestões.

Com o objetivo de fomentar a autonomia dos alunos e diminuir a sua dependência do professor, a ajuda à tarefa foi diminuindo gradualmente, à medida que se avançava nas tarefas. Esta diminuição gradual de orientação teve por base as orientações da metodologia *worked example*.

Após este período inicial de apresentação do projeto, os alunos começaram a resolver a tarefa 1, para tal fizeram o *download* do guião (anexo 2) disponibilizado na plataforma moodle.

Após fazerem o *download* foi pedido aos alunos que lessem o guião, que escolhessem depois o tema do seu projeto e, por fim, que criassem uma classe para agrupar os dados.

Em termos práticos para concretizar o projeto os alunos recorreram à tecnologia Microsoft Visual C# e criaram um novo projeto, ao qual adicionaram uma nova classe. Nessa classe tiveram de adicionar as variáveis (características do objeto) em função do tema que escolheram, e das características do objeto com que desejaram trabalhar. Tendo em conta as variáveis que escolheram, respeitando as regras da Programação Orientada a Objetos, relativamente ao encapsulamento de dados, tiveram de implementar os respetivos *gets* e *sets*. Também implementaram os construtores que acharam necessários.

Durante esta tarefa já fui dando poder de escolha e decisão aos alunos, permitindo-lhes, além de escolher o tema do projeto, a escolha do tipo de elementos que compõem a sua classe. Relativamente ao tema, como podemos verificar na seguinte tabela, metade dos alunos optaram por desenvolver o projeto adotando o tema fornecido a título de exemplo, e a outra metade escolheu outros temas de acordo com os seus interesses.

Aluno	Tema do projeto
1	Gestão de uma equipa de futebol
2	Gestão de uma equipa de futebol
3	Gestão de um Stand de automóveis
4	Gestão de uma equipa de futebol
5	Gestão de uma equipa de futebol
6	Gestão de uma loja de computadores
7	Gestão de um clube de vídeo
8	Gestão de uma equipa de futebol
9	Gestão de jogos
10	Gestão de um clube de vídeo
11	Gestão de uma frutaria
12	Gestão de uma equipa de futebol

Tabela 2 – Lista de temas escolhidos pelos alunos

Esta tarefa serviu como elo de ligação entre este módulo e os módulos restantes, pois permitiu aos alunos colocar em prática as competências adquiridas nos módulos anteriores e estabelecer nas próximas tarefas a conjugação dessas competências com os conteúdos deste módulo.

Em termos de conclusão sobre esta primeira tarefa do projeto, constatei novamente que os alunos apresentam muitas dificuldades ao nível não só da programação, como constatei com o teste diagnóstico, mas também ao nível dos conceitos fundamentais da Programação Orientada a Objetos, pois não sabiam adicionar uma nova classe a um projeto, não sabiam criar construtores nem encapsular dados. Mais uma vez esta verificação levou-me a reformular os problemas para as próximas aulas de maneira a rever estes conceitos. Optei por estas reformulações por estarmos a formar futuros técnicos que no próximo ano letivo vão estagiar na área da programação, numa empresa, e é essencial para que se mantenha o protocolo de estágios entre as empresas e a escola, que os alunos deixem uma boa imagem e que durante a sua passagem acrescentem valor à empresa. Hoje em dia um bom programador tem de dominar a Programação Orientada a Objetos.

Relativamente à opção de dividir o projeto em tarefas, considero até ao momento ter sido um sucesso, dado que todos os alunos concretizaram a tarefa 1 do projeto. Estes resultados começaram a fortalecer a minha opinião sobre a divisão do projeto em pequenas tarefas como um dos caminhos para o êxito do processo de ensino-aprendizagem da aprendizagem da Programação Orientada a Objetos em cursos profissionais.

Criação de uma conta na Dropbox.

Concluída a tarefa 1 informei os alunos que tinham de criar uma conta na *Dropbox* para guardarem os seus trabalhos e diários reflexivos.

Inicialmente os alunos demonstraram alguma resistência na criação da conta, refutando que não precisavam dela para nada, mas a verdade é que ao longo das aulas foram-se apercebendo da importância da mesma, referindo por diversas vezes “ainda bem que tenho uma cópia na *Dropbox*”. Isto acontecia essencialmente quando se esqueciam da *pen* em casa, ou quando alguém de outra turma, visto os computadores serem partilhados, apagava o seu trabalho.

Preenchimento do diário de aprendizagem

Ao faltarem 15 minutos para o final da aula, informei os alunos que deviam preencher o diário de aprendizagem relativo às duas aulas. Para o efeito criei e disponibilizei no *moodle* um documento

(anexo 6) para preencherem sobre o que aprenderam, o que fizeram e as dificuldades que encontraram durante as aulas. Após o seu preenchimento submeteram-nos no *moodle*.

2.4.2 Análise, compreensão e avaliação da ação

Da análise dos diários de aprendizagem deparei que os alunos têm consciência das dificuldades que têm ao nível dos ciclos e da criação de classes. Recorrendo à observação da aula, registada no diário de bordo e na grelha de observações aferi que os alunos participaram com entusiasmo em todas as atividades, criando sempre um diálogo construtivo. Também observei que os alunos compreenderam o conceito de stream e aplicaram-no corretamente nos casos práticos.

Foi crucial para o sucesso desta intervenção, a realização do exercício diagnóstico, a leitura e análise dos diários de aprendizagem dos alunos e a proximidade com os mesmos, para verificar que estes não tinham pré-requisitos para este módulo. Segundo o paradigma construtivista, o professor deve estimular o aluno a aprender partindo dos seus conhecimentos prévios, relacionando-os com os novos conteúdos, para que assim, mais facilmente possa atribuir um novo significado às novas aprendizagens. A verdade é que detetei que os conhecimentos prévios ao nível da programação praticamente não existiam, ou seja, o nível de conhecimentos estava muito afastado do esperado para este módulo. Refletindo sobre o assunto decidi ajustar as atividades ao nível dos alunos, e invocando aqui Lenine, decidi dar um passo atrás para mais tarde poder dar dois em frente.

Optei por adequar o nível, não simplificando, mas incorporando às aprendizagens do módulo outras aprendizagens que os alunos já deviam ter adquirido. Alguns alunos encontravam-se desmotivados a esta disciplina, devido ao fosso que existia entre aquilo que eles sabiam e o que lhes era exigido. Esta diferença era mais que suficiente para desistirem de realizar as aprendizagens devido à frustração de o não conseguirem. Para eliminar esta diferença, todas as apresentações, exercícios, problemas e projeto foram concebidos por mim, adaptando assim os conteúdos à estrutura de conhecimento dos alunos.

O uso de analogias foi uma aposta conseguida, através do mapeamento de similaridades os alunos facilmente conseguiram compreender os processos complexos associados ao manuseamento de *streams*. Esta conclusão é corroborada pelo estudo de Ferraz e Terrazzan (2002) no qual reconhecem o papel fundamental do uso de analogias no trabalho docente, reafirmando o seu uso como um dos recursos didáticos que norteará o trabalho do professor.

2.4.3 Aulas 3 e 4

Durante estas aulas os alunos realizaram uma série de atividades tendo em vista a construção e manipulação do conteúdo ficheiros de texto.

Análise colaborativa de mapas mentais para recapitulação/ligação de conceitos

A aula teve início com a contextualização dos alunos sobre os assuntos abordar, recorrendo a um mapa mental, com o qual estabelecemos a ligação entre os conteúdos das últimas aulas e os desta. Solicitei então aos alunos que relativamente às *streams*, relembassem os conceitos e as suas relações presentes no mapa mental (figura 21). De seguida, para concluir a análise deste mapa mental, falámos sobre a manipulação de ficheiros de texto, partindo dos conhecimentos adquiridos na aula anterior.



Figura 21 - Mapa mental 1 utilizado nas aulas 3 e 4

Mantendo o ambiente de diálogo/discussão que estava na sala, introduzi a leitura e escrita em ficheiros de texto, analisando com os alunos o código de uma função que permite a leitura de ficheiros de texto recorrendo ao método *readline()* da classe *StreamReader()*.

Dando continuidade à aula mostrei o código responsável pela manipulação de diretorias, conteúdo este que não pertence ao módulo lecionado, mas que achei relevante mencionar, pois as diretorias são uma ajuda na organização dos ficheiros.

Para terminar esta parte mais teórica, foi feita uma recapitulação da matéria dada nas aulas anteriores, uma síntese da aula de hoje e a antevisão da próxima aula tendo por base o seguinte mapa

mental. Sendo sempre os alunos a parte ativa deste processo de ensino-aprendizagem, de forma colaborativa foram explicando o diagrama e os vários conceitos a ele associados.

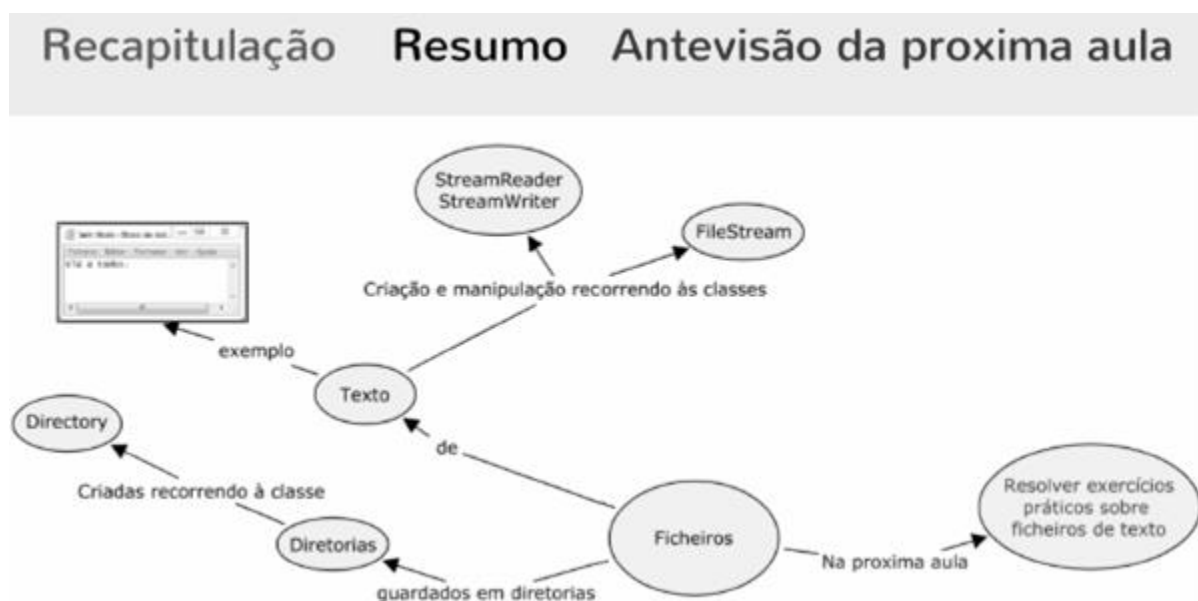


Figura 22 - Mapa mental 2 utilizado no final das aulas

Resolução do problema “Nomes e apelidos” - Problem-based learning

Iniciei esta atividade com a apresentação de um problema e solicitei aos alunos que construíssem uma aplicação em C# recorrendo à plataforma Microsoft Visual C#, como solução do mesmo.

Decidi aproximar o enunciado do problema à realidade dos alunos, referindo “na empresa onde vocês trabalham”, tentando assim motivá-los para a resolução do mesmo e preparando-os para o seu futuro profissional. Também tive o cuidado, aquando do desenho, de conceber um problema que levasse os alunos, durante a construção da solução, a rever e manipular conhecimentos como a construção de classes e manipulação de controlos, uma vez que demonstraram ter bastantes dificuldades nesses conteúdos.

Para além destes conhecimentos prévios para construírem uma solução, os alunos tiveram de mobilizar os conhecimentos adquiridos em aulas anteriores, tais como a manipulação de streams, e relacioná-los com os lecionados durante esta aula, ficheiros de texto, no sentido de encontrar uma solução para o problema.

Um desafio para vocês

Na empresa onde vocês trabalham pedem-vos uma aplicação que receba o nome e apelido de pessoas. A aplicação deve guardar o nome e o apelido das pessoas num ficheiro de texto e permitir ao utilizador pesquisar um apelido pelo nome.

Figura 23 - Problema “nomes e apelidos”

Relativamente ao desempenho dos alunos no problema, observei que não conseguiram concretizar uma solução logo à primeira, facto que não me preocupou, pois é perfeitamente normal e espectável que durante a construção de soluções para problemas, os alunos errem, testem diferentes soluções, reestruturem a sua solução, peçam a minha opinião, pois só com este envolvimento, participação ativa, tentativa/erro e reestruturação, conseguirão criar o seu próprio conhecimento.

Projeto - Resolução da tarefa 2.

Depois de terminarem os primeiros 90 minutos (aula 3), iniciei os segundos 90 minutos (aula 4), com a apresentação da segunda tarefa do projeto. Esta tarefa foi desenhada para permitir aos alunos, pegar no projeto desenvolvido na tarefa 1 e melhorarem-no acrescentando novas funcionalidades. Assim puderam concretizar e consolidar as suas aprendizagens, e desenvolver a flexibilidade cognitiva necessária através da aplicação dos mesmos conteúdos em situações diferentes.

Com o objetivo de desenvolver um sistema de login para a sua aplicação, os alunos tiveram que recorrer às aprendizagens realizadas nestas aulas e nas anteriores, nomeadamente sobre streams e ficheiros de texto, não esquecendo outros conteúdos/conhecimentos desenvolvidos ao longo do curso, como funções e ciclos.

Como aconteceu com a tarefa 1, também foi disponibilizado um guião para a tarefa 2 (anexo 3), no qual constavam algumas sugestões. Estas diziam respeito essencialmente a assuntos que os alunos precisavam de rever, exigiam raciocínio e esforço do aluno para entender e compreender a sua aplicabilidade, ou seja, o que o código fazia e onde devia ser aplicado. Este guião é uma estratégia da integração das metodologias da Dinamarca, na metodologia de aprendizagem baseada em projetos, com o objetivo de gradualmente se ir desvanecendo a orientação, promovendo a autonomia.

Preenchimento do diário de aprendizagem.

Os alunos preencheram o diário de aprendizagem relativo a estas aulas.

2.4.4 Análise, compreensão e avaliação da ação

Pela análise dos diários de aprendizagem constatei que os alunos têm dificuldades na conceção de funções. Tendo em atenção este facto, tive o cuidado de inserir nas atividades seguintes a manipulação de funções, para assim os ajudar a ultrapassar esta dificuldade, quer investigando na internet, quer praticando na plataforma Microsoft Visual C#.

Recorrendo à observação, mais uma vez constatei que os alunos estiveram participativos fazendo intervenções relevantes. Tendo em conta as características específicas de cada aluno, todos os problemas e dificuldades, vi este interesse e empenho como um passo importante, para a sua reconstrução enquanto alunos.

A turma voltou a referir que gostava da estrutura e tipo de apresentação eletrónica *Prezi*. Penso que este gosto pela apresentação, deve-se por um lado, ao facto da própria tecnologia ser apelativa, através das transições em *zoom*, e por outro lado, ao cuidado que tive na construção da aplicação, com esquemas, código e imagens que apelassem a participação dos alunos, por acreditar que a passividade acaba por saturar, abrindo a porta a assuntos divergentes.

Apesar de se verificarem melhorias óbvias, os alunos continuam a demonstrar alguma resistência quando se deparam com problemas mais complexos, que os obrigam a pensar, continuando a solicitar demasiado o apoio do professor. Decidi repensar no meu papel como orientador e, sempre que os alunos pediam a minha ajuda, eu deslocava-me junto deles e sugeria que pensassem mais algum tempo no problema, e que, caso fosse necessário, procurassem informação nos materiais que lhes facultei ou na internet. Procurei também valorizar quem conseguia resolver os problemas de forma autónoma, deslocando-me ao lugar, pedindo ao aluno que explicasse como ultrapassou o problema, o que fez, como fez, demonstrando que me preocupava e reconhecia o seu esforço, incentivando-os com expressões como “muito bem” ou “bom trabalho”.

Face ao esforço reduzido e pouca vontade de resolver situações que impliquem um raciocínio mais elaborado, procurei contrariar esta tendência, introduzindo em cada novo problema um desafio que os obrigasse a pensar e a desenvolver o pensamento computacional.

Apesar das dificuldades, os alunos conseguiram concretizar todas as atividades propostas para as duas aulas. Nos primeiros 90 minutos, resolveram com sucesso o problema que lhes apresentei e ainda parte do problema planeado para o dia seguinte. Sendo assim, ganhamos algum tempo para a

aula seguinte e, visto que os alunos ainda apresentavam muitas dificuldades ao nível dos ciclos e condições, decidi aproveitar esse tempo para lhes dar uma breve explicação sobre funcionamento dos mesmos.

Tentei sempre que os alunos desenvolvessem a sua flexibilidade cognitiva, para que ao longo da sua vida profissional, facilmente mobilizem os conhecimentos adquiridos durante a sua formação, na resolução dos mais diversos problemas. Um exemplo do desenvolvimento desta capacidade está na utilização de streams em diversos contextos, em exercícios, na resolução de problemas e por fim no projeto.

2.4.5 Aulas 5 e 6

As aulas 5 e 6 deram continuidade às atividades iniciadas durante as aulas 3 e 4.

Resolução do problema “Ficheiros de texto”-Problem-based learning

Inicialmente os alunos concluíram o problema que tinham iniciado na última aula (figura 24). Depois como já referi utilizei o tempo que sobrou para relembrar os ciclos. Recorrendo à caneta e ao quadro branco, fui lançando pequenos desafios, que depois em conjunto fomos resolvendo. Sempre que necessário explicava o funcionamento do ciclo.

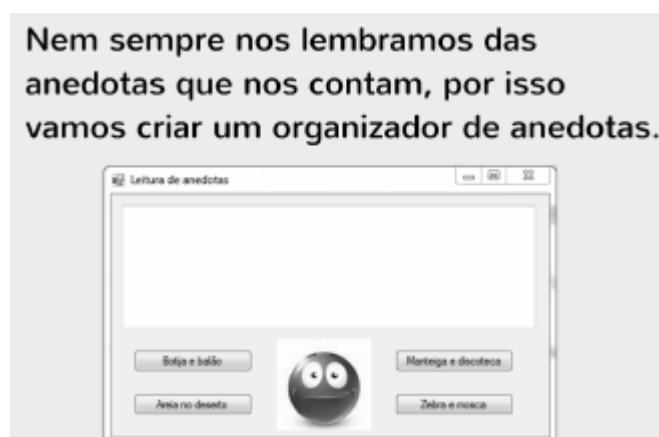


Figura 24 - Problema das anedotas

Projeto - Resolução da tarefa 2 (conclusão)

Na aula 6 os alunos continuaram a desenvolver a tarefa 2, reestruturando e expandindo o seu projeto.

Preenchimento do diário de aprendizagem.

O diário de aprendizagem foi preenchido como sempre no final da aula.

2.4.6 Análise, compreensão e avaliação da ação

Durante a primeira aula, os alunos concluíram o segundo problema com sucesso e no tempo que restou, como planeado, suprimiram algumas das suas dificuldades relativamente aos ciclos.

Na segunda aula, dedicada à conclusão da segunda tarefa do projeto, os alunos estavam muito agitados, sempre a trocar as janelas do computador, e como estranhei a situação, decidi perguntar o que se passava, nessa altura fui informado que o professor orientador decidiu, no meio da minha intervenção, estipular aquele dia como prazo final de entrega dos projetos relativos aos módulos anteriores à minha intervenção.

Esta situação acabou por desconcentrar os alunos da resolução da tarefa 2. Face a esta situação, e para não perder a atenção dos alunos por completo, tive de negociar com eles o tempo que tinham para concluir esses projetos e para trabalhar na tarefa 2. Isto porque com tanta confusão nem realizavam a tarefa nem concluíam os projetos anteriores. Decidimos então que metade da aula seria utilizada para concluir os trabalhos dos outros módulos e o tempo restante para concluir a tarefa 2 do projeto.

Como só tiveram disponível metade da aula, os alunos não conseguiram terminar a tarefa.

2.4.7 Aulas 7 e 8

Ao longo destas aulas, apresentei aos alunos um conjunto de atividades tendo em vista a construção e reconstrução de conhecimentos relativos aos conteúdos a lecionar, ficheiros binários e coleção *dictionary*.

Mesmo não fazendo parte deste módulo, decidi rever o conteúdo *dictionary*, por ser uma das coleções de dados mais importantes e por ter constatado ao longo do período de observação da prática pedagógica, que quando foram lecionadas as coleções, não foi dada a devida importância a esta classe e que, quando foi resolvido num dos módulos anteriores o projeto de criação de uma agenda, os alunos optaram por usar listas, quando deviam ter utilizado a coleção *dictionary*.

Assim, ao longo desta aula foram revistos e aprofundados os conteúdos relativos à coleção *dictionary*.

Como já foi referido anteriormente, todas as aulas tiveram como suporte a apresentação eletrónica em *Prezi*. A apresentação utilizada foi sempre a mesma, sendo redimensionada a cada nova aula. Optei por utilizar sempre a mesma apresentação *prezi*, porque assim tem a vantagem de permitir

regressar facilmente a momentos das aulas anteriores para fazer revisões. Assim sendo, nesta aula estabeleci a ligação às aulas anteriores e permiti aos alunos recordar conteúdos relacionados com as *streams*, importantes para esta aula. Foram então recordadas as classes de *stream*, *binaryWriter* e *binaryReader*, necessárias à escrita e leitura em ficheiros binários.

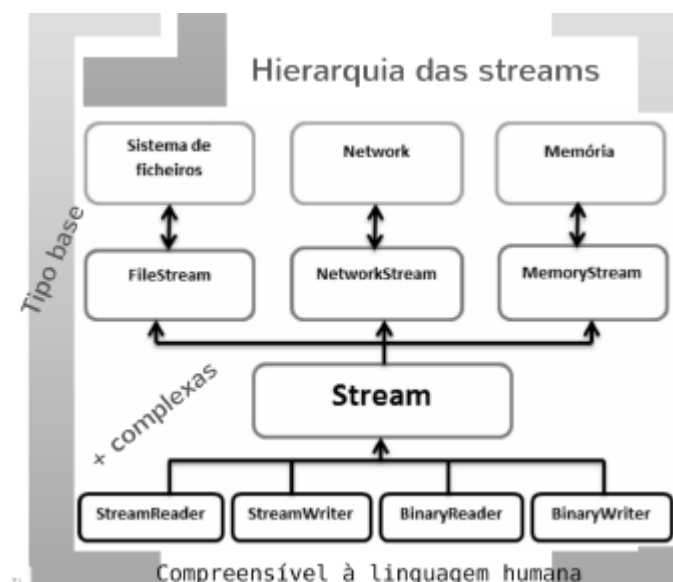


Figura 25 - Hierarquia de streams

De seguida falei do conceito de coleção, relativamente à coleção *dictionary*. Para facilitar a sua compreensão dei o exemplo de uma coleção de pessoas com características diferentes. Como tinha observado na primeira aula, os alunos tinham dificuldades em perceber o que são atributos de classes, como tal recorri a estas características para lhes explicar melhor o conceito de atributos. E assim com um exemplo apenas, resolvi as duas questões, o que é uma coleção e o que são os atributos de uma classe.



Figura 26 - Conceito de coleção

Para facilitar a aprendizagem coloquei no *Prezi* uma imagem (figura 26), que mostra um grupo de pessoas que possuem características diferentes. Através deste exemplo, os alunos verificaram que existem determinadas características que distinguem as pessoas umas das outras, concluindo assim que os atributos que caracterizam o objeto pessoa podem ser por exemplo a cor, a altura e a mobilidade. Através desta relação os alunos verificaram e compreenderam, a necessidade dos atributos nas classes, e que os mesmos servem para caracterizar os objetos.

Atividade colaborativa - quadro interativo (ficheiros binários).

Apresentei de seguida uma nova atividade, convidando os alunos a construir o seu próprio conhecimento através da troca de conhecimentos entre pares. Os alunos tiveram de comentar dois trechos de código. Um aluno foi ao quadro e, com a ajuda dos colegas, comentou o código e indicou se estávamos na presença de uma operação de leitura ou escrita em ficheiros. Este processo realizou-se para uma operação de leitura e para uma operação de escrita. Esta atividade utilizou como recurso o quadro interativo.



Figura 27 - Código apresentado aos alunos para comentarem

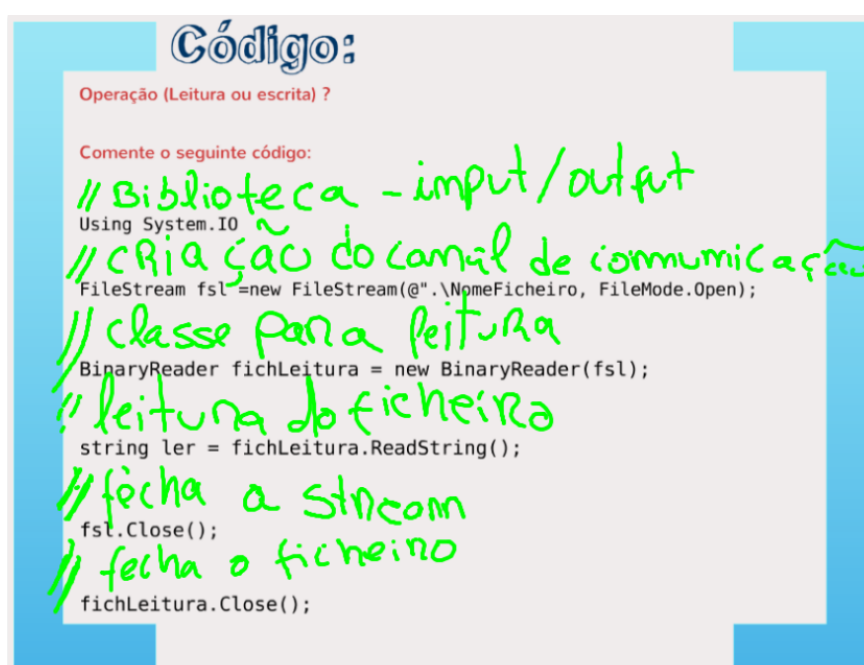


Figura 28 - Exemplo de código comentado pelos alunos

Esta atividade ao ser resolvida no quadro interativo despertou o interesse dos alunos. Estes estiveram atentos e participativos, verificando-se um sentimento de entreajuda ao comentar o código, criando assim um diálogo construtivo muito interessante. O sucesso desta atividade comprova que os alunos conseguiram concretizar com sucesso as aprendizagens anteriores e colocá-las em prática no momento certo. Não tiveram dificuldades em verificar se estávamos na presença de uma operação de leitura ou escrita em ficheiros binários

Iniciei esta atividade com a apresentação de um problema (figura 29), aos alunos para praticarem os conhecimentos relativos à coleção *dictionary*.

Com o objetivo de fazer uma pequena incursão nos domínios do construtivismo social, sugeri que, durante esta atividade, vestissem a pele de uma equipa de programadores, e trocassem informações, ideias e soluções entre si.

Desafio

Somos colaboradores da empresa de software "fam@soft", eu sou o gestor de projeto e vocês a equipa de programadores. Uma empresa de Call Center estabelece contato com a nossa empresa, para construirmos uma aplicação que:

- Sempre que surja um novo cliente, seja possível guardar o seu número de telefone, nome e localidade.
- Sempre que necessário se obtenha o nome e localidade de um determinado cliente, com base no seu número de telefone.




Figura 29 - Atividade colaborativa do problema do Call Center

Relativamente a esta atividade, os alunos empenharam-se na sua resolução. A forma como formulei o enunciado, dizendo que era um desafio e referindo que eu sou o gestor do projeto e eles os programadores, criou um ambiente de aprendizagem, que os motivou, induzindo-lhes uma vontade de conseguir resolver o problema de forma autónoma.

Como já referi anteriormente, estes alunos eram muito dependentes do professor, e tinham uma grande aversão ao ato de raciocinar, para começarem a ser mais persistentes e a pensarem por si, não lhes forneci pistas e deixei que eles tentassem resolver a atividade sozinhos.

Permiti-lhes assim realizar as aprendizagens, através da necessidade dos elementos, ou seja, ao construírem a solução do problema, os alunos foram-se apercebendo que precisavam de mobilizar

ou adquirir novos conhecimentos. Para tal passaram por um processo de procura, investigação, reflexão e teste.

Por exemplo, quando precisaram de ligar o cliente aos atributos telefone, nome e localidade, pesquisaram, refletiram, testaram e chegaram à conclusão que tinham de criar uma classe. Concretizada esta aprendizagem, quando no futuro se depararem com uma nova situação e precisarem de associar um objeto aos seus atributos, já saberão que necessitam de criar uma classe e como o fazer. São situações como esta que mostram a riqueza da aprendizagem através da resolução de problemas.

Outro exemplo, da construção de aprendizagens por descoberta, aconteceu quando precisaram de agrupar todos os clientes, logo tiveram de pensar em algo que lhes permitisse agrupar tais objetos. Chegaram à conclusão que precisavam de utilizar uma coleção, mas como também tinham de pesquisar elementos, optaram, corretamente, por utilizar a coleção *dictionary*.

Este tipo de aprendizagens é fundamental na aprendizagem de linguagens de programação, pois não chega que os alunos dominem a sintaxe e a funcionalidade dos vários elementos da linguagem, é essencial que também consigam estabelecer a ligação entre os mesmos e a sua utilização na resolução de diversos problemas. Só desta forma eles compreendem realmente o significado das aprendizagens.

A utilização deste problema, associado a toda a envolvimento criada, foi mais um passo para os alunos pensarem por si, compreenderem os conteúdos e reduzirem a sua dependência do professor. Esta capacidade de adaptarem os conhecimentos adquiridos, a uma nova situação, melhora e demonstra a flexibilidade cognitiva dos alunos.

A desvantagem desta atividade, foi a dificuldade de acompanhar a progressão dos alunos e tentar perceber quem fez o quê, porque facilmente em ambientes de aprendizagem cooperativos e colaborativos, os alunos ocultam as suas dificuldades por trás do trabalho dos colegas.

Projeto - Resolução da tarefa 3

Concluída a resolução do problema, dei início à aula seguinte com a apresentação da tarefa 3 do projeto. Como aconteceu com as outras tarefas, os alunos fizeram o *download* do guião que estava no moodle, e continuaram a ampliar e reestruturar o seu projeto. Apesar de já ter apresentado a tarefa 3 (anexo 4), devido ao incidente da aula anterior os alunos ainda tiveram de terminar a tarefa 2.

Os alunos preencheram e submeteram novamente os diários de aprendizagem no moodle.

2.4.8 Análise, compreensão e avaliação da ação

Com base nos registos de observação verifiquei que alguns alunos ainda apresentavam dificuldades ao nível do domínio da programação, tenho a consciência que programar para estes alunos, é um processo extremamente complicado, tendo em conta as suas características e falta de conhecimentos básicos. É de louvar que, apesar de todas estas barreiras, eles não desistiram e com maior ou menor ajuda, têm feito enormes progressos.

Programar é um processo extremamente complexo, exige que tenhamos a capacidade de fornecer ao computador um conjunto de instruções que resolvam um determinado problema. Temos de ter a habilidade de analisar o problema e desenhar uma solução. Esta fase é muito difícil para alunos com esta maturidade cognitiva pois, muitas vezes, exige deles um pensamento abstrato e matemático, que ainda não desenvolveram.

Depois segue-se a fase da transcrição da solução para uma linguagem que o computador entenda. Esta fase implica o domínio da sintaxe de uma linguagem de programação, algoritmia e, neste caso, do paradigma de Programação Orientada a Objetos.

Como os alunos apresentavam inúmeras dificuldades ao nível da programação, e este era o último módulo de programação, concluí que se não tentasse esclarecer agora as suas dúvidas, não existiria outra oportunidade. Após refletir no assunto, mudei a ação, abdicando do diário de aprendizagem e utilizando os 15 minutos que lhe estavam dedicados para o desenvolvimento do projeto. Tive de estabelecer prioridades, e para mim neste ponto era mais importante que os alunos realizassem determinadas aprendizagens.

Ao reduzir a ajuda aos alunos, quer ao nível do meu apoio para solucionar problemas, quer ao nível de sugestões no guião do projeto, com o intuito de fomentar a sua autonomia, reparei que a tendência para dispersarem para assuntos que não diziam respeito à aula era maior. Perante a frustração de não conseguirem progredir facilmente, porque existem obstáculos que têm de ser ultrapassados e competências que tem de ser adquiridas, os alunos estiveram menos motivados, e pioraram um pouco em termos de comportamento, mas nada que não se esperasse nesta fase de transição.

2.4.9 Aulas 9 e 10

Assim como aconteceu com a coleção *dictionary*, também me apercebi que os alunos, não possuíam conhecimentos sobre o controlo *listview* e, por considerar a sua importância decidi dedicarlhe esta aula.

Para contextualizar esse controlo com os assuntos precedentes apresentei um mapa mental (figura 30). Os alunos lembraram que uma classe permite criar objetos, que esses objetos podem ser guardados temporariamente e manipulados através da coleção *dictionary*, e que o controlo que vão aprender ao longo desta aula permite-lhes apresentar os objetos guardados na coleção.

Ao demonstrar nesta fase inicial a relação entre o conteúdo que vai ser aprendido nesta aula, e aqueles que os alunos já conhecem, pretendi dar um significado ao conteúdo, indicando a razão pela qual deve ser aprendido.

Ao longo deste projeto recorri constantemente a mapas mentais por concordar que “são uma ferramenta metacognitiva que por sua natureza gráfica podem tornar as informações desejadas mais acessíveis, o que por certo os credencia como uma importante ferramenta para expressão de conhecimento” (Gava, Menezes, & Cury, 2003, p. 3).

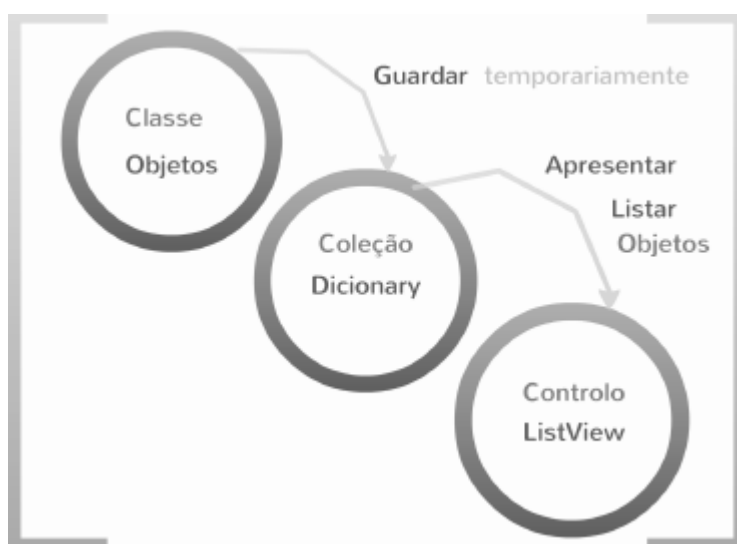


Figura 30 - Mapa mental para relacionar a aula com os assuntos precedentes

Os alunos perceberam a importância do controlo, e mencionaram que se o tivessem aprendido mais cedo, já o teriam utilizado na resolução de trabalhos de outros módulos. Este comentário comprovou a importância deste conteúdo e a sua relevância no panorama da programação.

Atividade Desafio Call Center upgrade (ListView) – problema-based learning

Durante esta atividade os alunos resolveram um problema para concretizarem e consolidarem aprendizagens sobre *listViews*. Solicitei-lhes que recorressem à solução que desenvolveram na aula anterior e aumentassem a sua complexidade e funcionalidades recorrendo aos conhecimentos que aprenderam nesta aula. A reestruturação da aplicação desenvolvida na aula anterior permitiu que esta listasse todos os clientes do *Call Center*.

Projeto - Resolução da tarefa 3 (conclusão)

Os alunos concluíram a tarefa 3 e, mobilizando os conhecimentos que adquiriram sobre o controlo *ListView*, adicionaram ao seu projeto a funcionalidade de listar todos os elementos contidos na classe *dictionary*.

Ao aplicarem o conhecimento sobre o controlo *ListView*, em contextos diferentes os alunos estão a desenvolver a flexibilidade cognitiva. Segundo Spiro e Jehng (1990), para desenvolverem a flexibilidade cognitiva os alunos devem ser capazes de reestruturar o seu conhecimento para resolver as situações com que se deparam.

2.4.10 Análise, compreensão e avaliação da ação

As aulas correram bem, os alunos mais uma vez estiveram participativos, sempre com intervenções e questões pertinentes. Todos conseguiram resolver o problema de *upgrade* ao desafio do *Call Center*, acrescentando uma *listview* para mostrar e remover clientes.

Observei que os alunos se estão a tornar mais autónomos na resolução dos seus problemas, apoiando-se no material fornecido e na internet.

Os alunos já começam a associar o que se dá na aula anterior ao que estão a fazer no projeto. O progresso das suas aprendizagens é notório na medida em que já conseguem adaptar o conhecimento adquirido a novas situações. O facto de o projeto ter ido ao encontro dos seus interesses, ou seja, ao tema que eles escolheram, contribuiu para a sua motivação na concretização do mesmo. Esta metodologia de projeto resultou muito bem, porque deu margem de manobra para os alunos aprenderem e relembrarem matérias que vão além dos conteúdos pré estabelecidos para este módulo.

Ao se concentrarem mais no processo de ensino-aprendizagem os alunos melhoraram significativamente o seu comportamento, reduzindo o tempo que passavam na internet com assuntos externos à aula.

Os alunos trabalharam a ritmos diferentes no projeto, alguns alunos ficaram mais atrasados e não conseguiram completar a tarefa 3. Facto que não me preocupou muito, pois uma das vantagens desta metodologia é permitir ritmos de aprendizagem diferentes.

2.4.11 Aulas 11 e 12

Ao longo destas aulas procurei que os alunos realizassem aprendizagens sobre o processo de serialização de ficheiros binários.

Inicialmente, para despertar o interesse dos alunos e mostrar-lhes a importância do que iam aprender, apresentei um mapa mental (figura 31), o mesmo da aula anterior ao qual acrescentei o conceito a lecionar durante esta aula. Também recorri ao mapa para que os alunos entendessem a diferença entre o dicionário, que agrupa e guarda os dados temporariamente, e o ficheiro, que guarda os dados de forma permanente. O mesmo mapa serviu para compreenderem que, para guardar objetos de uma forma eficiente em ficheiros binários, devem recorrer ao processo de serialização e para reverterem o processo devem recorrer à desserialização do objeto.



Figura 31 - Mapa mental para relacionar conteúdos

Atividade construtivista de construção de conhecimento por analogia (sapo insuflável versus serialização)

Visto a serialização ser um conteúdo bastante complexo, recorri ao uso de uma analogia (figura 32) para ajudar os alunos a construírem e consolidarem os seus conhecimentos sobre o processo de serialização.

No início os alunos associaram os valores dos atributos de um objeto ao estado do mesmo, pois o que é guardado em ficheiro através do processo de serialização é o estado do objeto. Como podemos verificar na seguinte imagem, facilmente, eles associaram as características do sapo ao seu estado (ex.: cocas, verde e masculino).



Figura 32 - Analogia entre a serialização e o sapo insuflável

De seguida, para compreenderem o processo de serialização, os alunos fizeram a comparação baseada em similaridades entre as duas estruturas, a que já conheciam e o novo conhecimento a adquirir. Num ambiente de diálogo/discussão os alunos chegaram à conclusão que o sapo insuflável é o objeto que pretendemos serializar, que o ar é o seu estado e que o frasco é o ficheiro. Tendo estes elementos análogos em mente, compreenderam que quando um objeto é serializado, o seu estado (ar) é guardado em ficheiro (frasco) e para reverterem o processo, desserializar o objeto, o seu estado (ar) volta a ser carregado do ficheiro para o objeto.

Fornecendo assim conteúdos relacionados com a estrutura de conhecimentos dos alunos, com a sua interação com o meio, permiti que reestruturassem mentalmente as suas estruturas, concebendo para tal, novos esquemas para apreensão do novo conteúdo.

Deste modo, numa conceção construtivista, os alunos foram capazes de elaborar uma representação pessoal sobre o novo conteúdo, não partindo do vazio, mas sim de conhecimentos prévios ou experiências já conhecidas, os quais conseguiram relacionar com o novo conteúdo resultando na apropriação do mesmo.

Atividade Desafio Call Center upgrade (Serialização) – problema-based learning

Esta atividade permitiu aos alunos recorrerem ao problema que tinham desenvolvido ao longo das aulas anteriores e acrescentarem a funcionalidade de guardar os dados em ficheiros, através do processo de serialização.

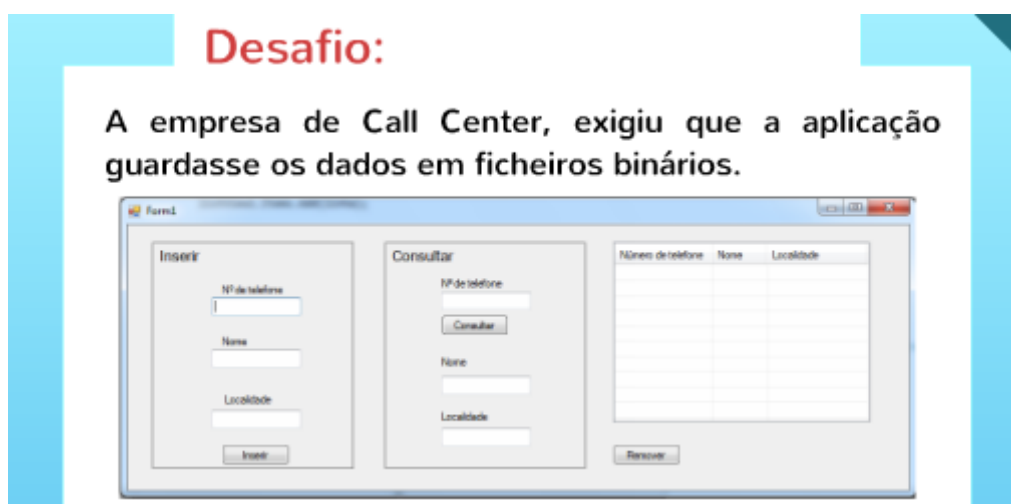


Figura 33 - Problema sobre serialização.

Projeto - Resolução da tarefa 4

Depois de concretizarem o problema, deu-se início aos segundos 90 minutos de aula. Informei os alunos que estava disponível no moodle o guião da tarefa 4 (anexo 5) relativa ao projeto. Esta tarefa permitiu aos alunos relacionar e aplicar os conteúdos apreendidos nos 90 minutos anteriores (serialização) a uma nova situação. Assim sendo, os alunos utilizaram o conceito de serialização e de desserialização para guardarem as suas coleções de elementos em ficheiros binários.

2.4.12 Análise, compreensão e avaliação da ação

Relativamente à resolução do projeto os alunos continuaram a desenvolvê-lo cada um ao seu ritmo. Como eram eles que geriam o seu tempo e aprendizagens o resultado era ritmos de trabalho diferentes.

2.4.13 Aulas 13 e 14

Nas aulas 13 e 14 os alunos aprenderam vários controlos de inserção de dados em formulários. Decidi fazer uma revisão aos diversos controlos por achar que seriam importantes para a conclusão dos projetos.

Projeto - Resolução da tarefa 4

Os alunos continuaram a trabalhar no seu projeto.

2.4.14 Análise, compreensão e avaliação da ação

Estas foram as últimas aulas que os alunos tiveram para concluir os seus projetos. Nem todos os alunos conseguiram concluir o seu projeto, mas tendo em atenção o cenário inicial, penso ser um sucesso a maioria ter conseguido.

No entanto, os alunos que não concluíram o projeto comprometeram-se a termina-lo em casa.

2.4.15 Aulas 15 e 16

De acordo com a planificação da intervenção (figura 34), as aulas 15 e 16 foram reservadas para os alunos fazerem uma primeira apresentação dos seus projetos à turma e receberem o feedback dos restantes elementos presentes.

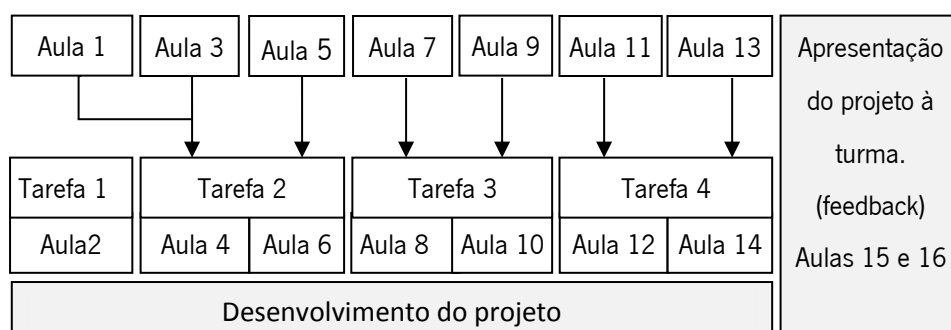


Figura 34 - Esquematização desde a conceção do projeto ao seu feedback

Inicialmente informei os alunos que dispunham de 10 minutos para apresentarem o seu projeto, e que depois, seria aberto um espaço para confronto/discussão de ideias e, os colegas, o professor orientador, eu e o meu colega de estágio faríamos observações e daríamos sugestões de melhorias relativamente à sua solução.

Por fim, os alunos foram informados que teriam a semana seguinte para, em casa, reformularem, caso fosse necessário, os seus projetos.

2.4.16 Análise, compreensão e avaliação da ação

Verifiquei através das apresentações, que 1 dos 5 alunos terminou o projeto em casa, e 2 progrediram bastante. Também constatei que alguns alunos que já tinham concluído o projeto, fizeram melhorias em casa, acrescentando funcionalidades e pormenores ao seu projeto que não tinham sido sugeridos. As duas situações demonstram a motivação e autonomia dos alunos, tanto pelo facto de por sua iniciativa trabalharem em casa, como pelo facto de, por sua vontade, irem além do solicitado.

Os alunos comportaram-se decentemente na apresentação, os colegas estiveram sempre atentos e participativos, com intervenções relevantes sugerindo melhorias. Durante este período, observei que os alunos davam especial importância ao que os colegas faziam de diferente, referindo muitas vezes, “essa parte está fixe, vou tentar fazer isso em casa”. As apresentações acabaram por despertar a vontade de alguns alunos fazerem mais.

2.4.17 Aulas 17 e 18

Após terem tido uma semana para reformularem os seus projetos os alunos apresentaram e defenderam a sua versão final do projeto, a qual avaliei recorrendo a uma grelha de avaliação construída para o efeito (anexo 8).

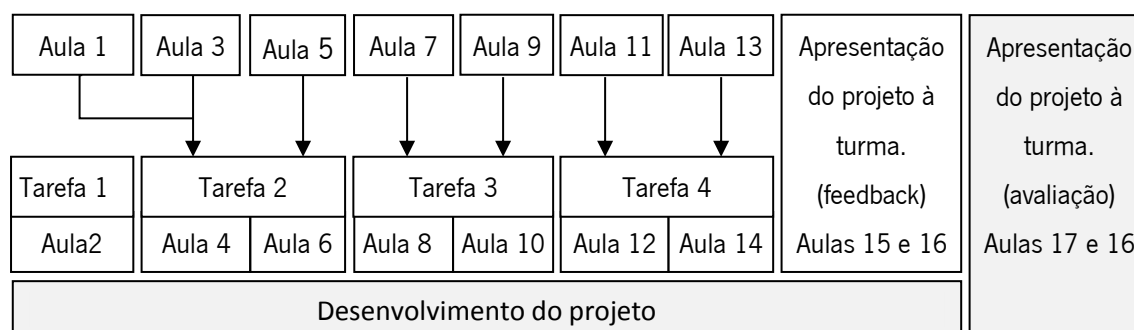


Figura 35 - Esquematização desde a conceção do projeto à sua avaliação

Os alunos tiveram, como aconteceu na última apresentação, 10 minutos para apresentarem o seu projeto. Terminada esta apresentação, o júri composto por mim, pelo meu colega de estágio, e pelo professor orientador, realizou uma série de questões, para validar o conhecimento que os alunos tinham da sua aplicação. No final era preenchida a grelha de avaliação.

2.4.18 Análise, compreensão e avaliação da ação

Ao realizar duas sessões de apresentações procurei criar oportunidades para os alunos desenvolverem a sua autoconfiança.

Todos os alunos que ainda não tinham concluído o projeto, fizeram-no durante a semana, e a maioria acrescentou pormenores e funcionalidades ao seu projeto.

De uma maneira geral os projetos estavam muito bons (anexo 10).

2.5 Avaliação do Projeto de intervenção

Após a implementação das diversas atividades, estratégias e metodologias já referidas, para desenvolvimento da autonomia, motivação e de um vasto conjunto de aprendizagens, torna-se necessário aferir os resultados obtidos e a evolução ocorrida ao longo do processo de intervenção. Assim, começarei por fazer a avaliação do projeto desenvolvido pelos alunos, de seguida a análise dos dados recolhidos através da grelha de avaliação de comportamentos, depois analisarei as respostas dos alunos ao questionário de avaliação da intervenção e, por fim, avaliarei a intervenção à luz dos seus objetivos.

2.5.3 Avaliação dos projetos desenvolvidos pelos alunos

Relativamente ao projeto desenvolvido pelos alunos ao longo desta intervenção, considero que foi um sucesso. A abordagem construtivista de aprendizagem baseada em projetos, associada à metodologia utilizada na Dinamarca, permitiu subdividir o projeto em pequenas tarefas que, à medida que iam sendo concretizadas, aumentavam as funcionalidades e a complexidade do projeto. Ao mesmo tempo foi sendo reduzida a orientação, diminuindo a dependência do professor, e promovendo assim a autonomia dos alunos. O projeto superou os objetivos esperados, na medida em que a maioria dos alunos, por iniciativa própria, conseguiu exceder o que foi pedido. Os alunos tiveram a destreza de adicionar, novos elementos e funcionalidades ao seu projeto de forma autónoma, muitas vezes em casa, indo além das quatro tarefas propostas. Eles próprios chegaram à conclusão que o seu projeto beneficiaria com a inclusão destas novas funcionalidades e objetos. Este facto permitiu a obtenção de soluções diferentes, enriquecendo as apresentações finais.

Este projeto foi avaliado tendo por base a grelha de avaliação do projeto (anexo 8) e é um dos principais elementos de avaliação do projeto de intervenção. Quantitativamente aplicando esta grelha, os projetos tiveram uma avaliação muito boa, sendo a média de 15 em 20 valores. A tabela seguinte mostra a avaliação individual dos projetos.

Aluno	Avaliação final do projeto
1	16
2	19
3	15
4	13
5	16
6	14
7	13
8	13
9	15
10	15
11	14
12	16

Tabela 3 – Avaliação individual do projeto desenvolvido pelos alunos

Tendo em consideração a complexidade do projeto desenvolvido, a quantidade de conteúdos abordados e o pouco tempo disponível, como mencionaram os alunos nas respostas ao questionário, que serão apresentadas nesta secção, não posso deixar de salientar os bons resultados obtidos e o contributo da metodologia de aprendizagem baseada em projetos para o sucesso das aprendizagens dos alunos, ajudando-os a desenvolver a confiança para aprender a pensar e a resolver.

2.5.4 Análise da progressão da autonomia, motivação e comportamento

No final de cada aula registava o desempenho dos alunos numa grelha de observações (anexo 8), criada para ao efeito.

Estes dados serviram como indicadores ao longo da intervenção e permitiram-me ajustar a minha ação. Para cada item da grelha, conforme o desempenho do aluno, colocava um dos seguintes indicadores:

Indicador	Valor de referência
(NC) Não concretizou	0
(CR) Concretizou Razoavelmente	0,5
(C) Concretizou	1

Tabela 4 – Indicadores do desempenho dos alunos

Para uma melhor compreensão dos dados recolhidos relativamente à autonomia, motivação e comportamento dos alunos, criei o seguinte gráfico.

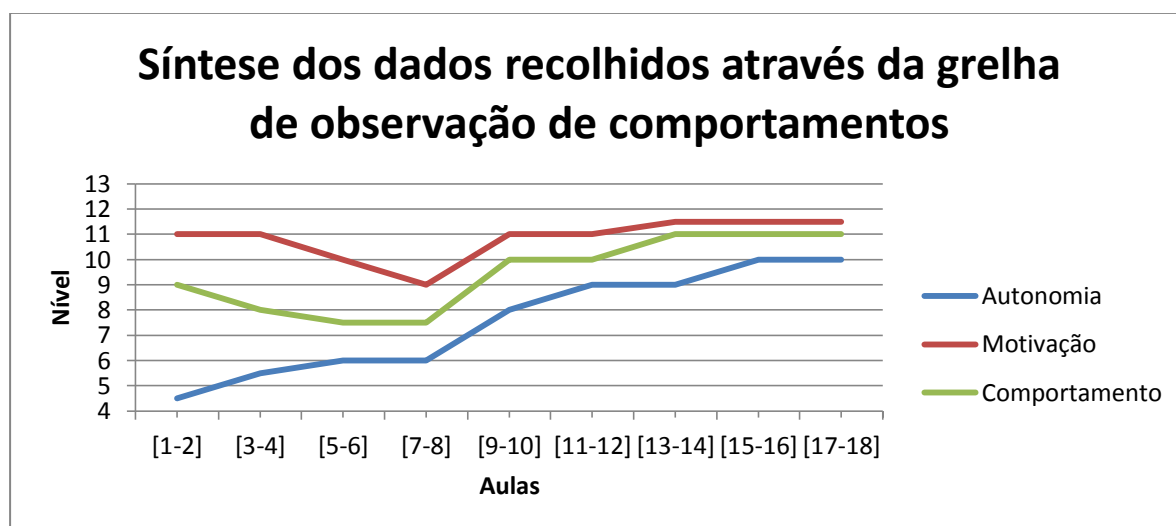


Gráfico 1 – Síntese dos dados relativos recolhidos relativamente à autonomia, motivação e comportamentos

Analisando os dados recolhidos e apresentados no gráfico, podemos verificar que inicialmente, na primeira aula os alunos estavam motivados e com um comportamento satisfatório, isto deveu-se ao fator novidade, um novo professor, novas metodologias e tecnologias. No entanto, no que diz respeito à autonomia, os alunos eram pouco autónomos.

É de salientar que os alunos estavam motivados, tinham vontade de fazer, mas faltava-lhes desenvolver hábitos, competências e métodos de trabalho para serem mais autónomos.

No intervalo de [1-8] os alunos a nível comportamental pioram, pois o professor e as tecnologias deixaram de ser novidade e as atividades exigiam uma participação ativa, para a qual eles ainda não estavam preparados. Todas estas razões, associadas ao facto de os alunos terem de terminar os projetos de outros módulos durante este período, e de eu ter reduzido a intensidade da minha ajuda para fomentar a autonomia, acabaram por se refletir na redução da sua motivação.

Dentro das linhas construtivistas, quando mudamos a nossa ação, normalmente o efeito no comportamento dos alunos não é imediato.

De acordo com os dados do intervalo [7-10], a autonomia, motivação e comportamento melhoraram significativamente, em primeiro lugar porque os alunos, começaram a perceber a dinâmica das metodologias e das aulas, em segundo lugar porque já começavam a adquirir habilidades que lhes permitiam ser mais autónomos e, em terceiro lugar, porque entraram numa fase onde os conteúdos eram mais agradáveis devido ao facto de serem mais visuais.

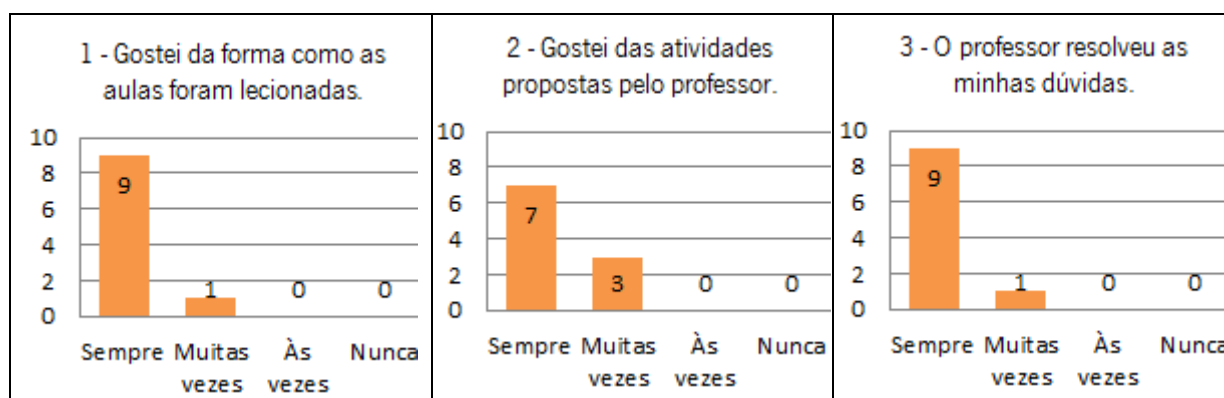
Na parte final, os alunos atingiram um nível de motivação, autonomia e comportamento bastante elevado, fruto da capacidade de concretização das atividades e da qualidade do seu projeto.

2.5.5 A opinião dos alunos

Os dados relativos à opinião dos alunos sobre o Projeto de Intervenção foram recolhidos através do método de inquérito e da técnica de questionário. Responderam a este questionário 10 dos 12 alunos.

O questionário (anexo 9) foi produzido recorrendo à tecnologia Google Drive e disponibilizado aos alunos sob a forma de hiperligação através da plataforma moodle.

Relativamente às questões fechadas a opinião dos alunos foi a seguinte:



Gráficos 2,3 e 4 – Opinião dos alunos sobre o projeto de intervenção

De acordo com as respostas dos alunos, podemos verificar que todos os alunos inquiridos gostaram da forma como as aulas foram lecionadas, das atividades e do apoio que lhes forneceu.

Relativamente às questões abertas a opinião foi a seguinte:

À questão 4 - Indica as coisas de que mais gostaste nestas aulas, os alunos responderam:

- *“O que mais gostei foi da maneira das aulas serem lecionadas com recurso a multimédia. A ajuda dada pelo professor também foi muito importante.”*
- *“Gostei da maneira como as aulas eram dadas e os projetos também eram divertidos.”*
- *“As apresentações, a forma de explicar os assuntos lecionados e a maneira de como o professor conseguiu cativar atenção dos alunos para o assunto tratado. As questões que efetuei foram bem resolvidas pelo professor. O modo como a aula se realizava era muito atrativa.”*
- *“O que mais gostei nestas aulas foi a maneira que o professor nuno apresentou as aulas. Foi bastante simples e ajudava imenso os alunos.”*
- *“A maneira de o Professor Nuno dar as aulas.”*
- *“Gostei de aprender coisas novas.”*
- *“Gostei da forma como as aulas foram lecionadas; e da maneira como o professor interagia connosco.”*
- *“A forma como a matéria foi explicada, e a realização do projeto.”*
- *“O que mais gostei de fazer nas aulas foi o projeto.”*

Quanto à questão 5 - Indica as coisas de que menos gostaste, as respostas foram:

- *“A coisa que menos gostei foi da exigência dos projetos.”*
- *“Não tenho nada a dizer.”*
- *“Sem nada a dizer...”*
- *“Acho que devia haver mais tempo.”*
- *“Gostei de tudo porque este módulo tem matéria que eu achei interessante.”*

Este projeto de intervenção foi concebido a pensar nos alunos, e por isso toda a reflexão, investigação e ação foi conduzida, não só no sentido de promover o sucesso das suas aprendizagens, mas também para colmatar as suas dificuldades e limitações.

Podemos constatar através das opiniões dos alunos, que estes gostaram da abordagem, das metodologias, dos recursos, das estratégias e das atividades que lhes apresentei ao longo desta intervenção. Verifica-se também que se aperceberam da relação de proximidade que mantive com eles,

estando sempre presente para os ajudar e orientar. Esta proximidade professor/aluno decorreu por dois motivos: primeiro porque faz parte da minha personalidade enquanto professor, e segundo porque a abordagem construtivista que adotei tanto proporciona uma maior comunicação nos dois sentidos, como liberta o professor para andar pela sala, ajudando e orientando os alunos sempre que necessário. Esta opinião dos alunos demonstrou a relevância da relação professor/aluno para o sucesso do processo de ensino-aprendizagem.

De notar também que os alunos reconheceram a importância da apresentação eletrónica Prezi como suporte às aulas. Esta admiração por esta tecnologia surge do facto de esta ser novidade para eles, da harmonização na disposição de imagens, esquemas, texto, trechos de código tornando assim a apresentação apelativa e nada maçadora. Outro aspeto que cativou a sua atenção foi a apresentação ser desenhada para proporcionar a participação ativa dos alunos e, desta forma, distanciar-se das apresentações convencionais de exposição de matéria.

Constata-se também que os alunos gostaram de desenvolver o projeto, indicando inclusive que era divertido. Em muito contribuiu para esta opinião o facto de os projetos terem sido desenhados em função do seu nível, dificuldades e interesses.

2.5.6 Avaliação do projeto de intervenção à luz dos objetivos

(i) Avaliar o impacto da abordagem construtivista no ensino-aprendizagem da Programação Orientada a Objetos

Após um contacto mais próximo com os alunos, constatando as dificuldades dos mesmos ao nível das bases na Programação e Programação Orientada a Objetos, refleti e mudei a minha ação reformulando o primeiro objetivo do plano de intervenção para avaliar o impacto da abordagem construtivista no ensino-aprendizagem da Programação Orientada a Objetos.

Pude verificar com esta intervenção, que a abordagem construtivista se adequa e contribui para o sucesso do ensino-aprendizagem da Programação e da Programação Orientada a Objetos em cursos profissionais, facto verificado pelos resultados da avaliação do projeto desenvolvido pelos alunos, elemento que comprova tanto a concretização de aprendizagens como de competências.

Os alunos não só aprenderam e dominaram os conteúdos propostos pelo referencial, e depois transcritos para a planificação, as *streams* e os ficheiros, como também adquiriram e reaprenderam outros conteúdos como a coleção *dictionary*, a *listView* e diversos controlos, ao mesmo tempo que

desenvolviam o seu pensamento computacional, praticando e relembrando procedimentos como a criação de classes, funções, encapsulamento de dados e ciclos.

Da análise das grelhas de observação, de avaliação e dos diários de bordo, constata-se que em muito contribuíram as várias metodologias baseadas no paradigma construtivista, para o sucesso deste processo de ensino-aprendizagem. A utilização das metodologias de aprendizagem baseada em problemas e projetos permitiram envolver os alunos ativamente, na resolução de diversos casos práticos próximos do real. Durante a construção das soluções os alunos não só aprenderam a interligar os novos conhecimentos com os conhecimentos prévios, como também aprenderam a aplicar os conhecimentos adquiridos a novos contextos, desenvolvendo assim a flexibilidade cognitiva.

Tanto os problemas como o projeto desenvolveram o espírito investigativo e reflexivo dos alunos, ambos muito importantes no domínio da programação. Sempre que se deparavam com um problema, os alunos formulavam e testavam hipóteses, refletiam sobre os resultados, procuravam dados adicionais e reformulavam as hipóteses até encontrarem uma solução.

Como um dos grandes problemas era a inexistência de conhecimentos prévios em programação, existiu o cuidado de integrar a revisão de conhecimentos anteriores na elaboração dos problemas e projetos, permitindo assim a superação das dificuldades existentes ao nível da programação.

Fazendo aqui uma conclusão antecipada, constatei que o sucesso da utilização de problemas e projetos no ensino da programação depende em muito da capacidade do professor formular esses problemas e projetos. É necessário que este seja capaz de conhecer bem os seus alunos, as suas dificuldades e necessidades, para que assim elabore projetos e problemas que desenvolvam as habilidades e atitudes desejáveis.

Outra metodologia realmente importante neste processo de ensino-aprendizagem foi o uso de analogias, que permitiram a construção ativa do conhecimento fazendo a aproximação entre domínios familiares e outros que tinham que ser compreendidos ou aprendidos (Harrison & Treagust, 2006). Em termos práticos, o uso de analogias permitiu a compreensão e construção de aprendizagens sobre conceitos complexos de Programação Orientada a Objetos, que depois os alunos mais facilmente aplicaram na prática.

Sintetizando, o recurso a metodologias baseadas em pressupostos construtivistas deu aos alunos a oportunidade de compreender, de aprender para o uso, de desenvolver as habilidades do pensamento, refletir, reformular e assim efetuar aprendizagens e adquirir competências para a vida.

(ii) Verificar a adequação de alguns princípios didáticos de ensino da informática na Dinamarca ao nosso sistema de ensino profissional

Ao longo desta intervenção, constatei que as metodologias de ensino utilizadas no ensino da programação na Dinamarca, como a *stepwise improvement*, *worked example*, e o e-portfólio se adequaram perfeitamente, trazendo vantagens ao ensino da programação no nosso sistema de ensino profissional.

Analisando as metodologias de ensino da programação na Dinamarca, verifica-se que estas vão de encontro ao paradigma construtivista. São metodologias onde o aluno é o elemento ativo do processo de ensino-aprendizagem e é ele mesmo que vai construindo o seu projeto gradualmente. Por este facto estas metodologias facilmente se integraram nas metodologias construtivistas que estava a utilizar, dando o seu contributo para o sucesso do processo de ensino-aprendizagem.

A aplicação da metodologia *stepwise improvement*, divisão do projeto dos alunos em várias tarefas, permitiu a existência de várias metas, em vez de só uma, possibilitando assim que os alunos se autorregulassem mais facilmente, e mais facilmente recebessem um *feedback* do seu trabalho, fornecido pela tecnologia ou pelo professor.

O recurso à *worked example* permitiu desvanecer a orientação, fornecida pelo professor ao longo das tarefas, promovendo assim a autonomia. Tanto esta metodologia como a anterior permitiram gerir sentimentos de frustração e desânimo.

A utilização do e-portfólio possibilitou aos alunos ter sempre disponível o material necessário ao bom funcionamento das aulas. Como o desenvolvimento do projeto se realizou durante toda a intervenção, os alunos precisavam de ter sempre disponível o trabalho desenvolvido nas aulas anteriores. Uma vez que os computadores eram partilhados por outras turmas, os projetos acabavam por desaparecer e como os alunos nunca traziam *pen*, o e-portfólio foi a solução para este problema.

(iii) Analisar o contributo da abordagem construtivista e das metodologias de ensino da programação na Dinamarca na promoção da autonomia e motivação dos alunos.

Este objetivo surge da divisão do primeiro objetivo do plano de intervenção em dois, separando-se assim a aprendizagem da Programação Orientada a Objetos da promoção da autonomia e motivação dos alunos.

A integração dos princípios didáticos Dinamarqueses na abordagem construtivista permitiram criar ambientes de aprendizagem favoráveis à promoção da autonomia e da motivação. Tendo-se constatado inicialmente que os alunos estavam completamente dependentes do professor e da sua solução, foi bom verificar a sua evolução ao longo da intervenção (gráfico 1) e constatar que no final, já conseguiam pensar, resolver problemas e efetuar descobertas por si próprios. Criaram-se assim condições pedagógicas favoráveis ao desenvolvimento de capacidades que lhes permitiram realizar as aprendizagens baseadas em matérias de apoio e na pesquisa e não tanto no professor. Tendo por base o relatório de atividades do moodle (anexo 11), podemos verificar o elevado número de consultas que os alunos fizeram aos diversos documentos de apoio facultados na plataforma. Este facto comprova como estas estratégias pedagógicas estimularam os alunos a desenvolver o conhecimento pelos próprios meios.

Ao recorrer às metodologias de ensino da programação na Dinamarca, simplificou-se o processo de autorregulação dos alunos, pois fragmentar o projeto em várias tarefas, facilita tanto o controlo temporal sobre o projeto, como o controlo sobre as aprendizagens. Este facto permitiu-lhes irem concretizando o seu projeto, passo a passo, ultrapassando assim pequenas metas, concretizáveis por eles, alcançando assim o sucesso, e desta forma continuando motivados. Segundo Schunk (1991) “indivíduos que se sentem eficazes trabalham mais e persistem por mais tempo quando se deparam com dificuldades do que aqueles que duvidam das suas capacidades” (p. 208).

Note-se que a utilização de metodologias de aprendizagem baseadas em problemas e projetos melhora a motivação dos alunos através da simulação de casos práticos que vão de encontro aos interesses dos alunos. Da mesma forma, desenvolve a autonomia ao permitir que se envolvam na procura de uma solução, procurando respostas, refletindo, investigando e reformulando. A utilização destas metodologias permite melhores resultados quando relacionadas com as metodologias do ensino da informática na Dinamarca como a *worked example*. Recorrendo a esta metodologia, para não se criar frustração e desmotivação nos alunos, foi proporcionada alguma ajuda (“andaimes”) nas primeiras tarefas do projeto, através do guião e da minha ajuda, e com o decorrer da intervenção estes “andaimes” foram sendo retirados, salvaguardando-se assim a motivação e promovendo-se a autonomia.

Tendo adotado um papel mais de facilitador, orientador e consultor, levei os alunos a assumir a responsabilidade pela sua própria aprendizagem, encorajando a criatividade e iniciativa, e a aprender a aprender, e não menos importante, a aprender a fazer.

Os alunos evoluíram significativamente relativamente à autonomia e à motivação. O facto de estarem a fazer algo que gostam, que lhes dá prazer, construído por eles, que conseguem ver o significado e que vai de encontro aos seus interesses, permitiu-lhes mudar a sua postura perante as aulas

2.6 Dificuldades encontradas

Ao longo da intervenção, deparei-me com alguns obstáculos, podendo referir desde já a heterogeneidade e os graves problemas de comportamento desta turma.

A minha intervenção, que estava planificada para a interrupção letiva da faculdade e início do segundo semestre, teve que ser antecipada à última da hora, devido a uma alteração do calendário modular efetuada pelos professores responsáveis pela disciplina, para o final do primeiro semestre, ou seja, fase de maior trabalho na Universidade. Como o projeto estava desenhado para o ensino da programação, e este módulo era o último de programação, não deu para escolher outro. Esta mudança implicou uma maior concentração de trabalho, uma vez que estávamos no final do primeiro semestre, altura de concluir e entregar os trabalhos na Universidade.

Como já referi, o módulo lecionado, é o último de programação, onde são lecionados os conteúdos de Programação Orientada a Objetos Avançada, ou seja, alguns dos conteúdos mais complexos.

Foi a primeira vez que trabalhei com a linguagem de programação C# e, como lecionei o último módulo, e os módulos têm uma continuidade, tive que aprender quase a totalidade da sintaxe desta linguagem, pois só assim poderia orientar e acompanhar devidamente os alunos.

Como as disciplinas técnicas geralmente não têm manuais, fui obrigado a elaborar todos os materiais pedagógicos utilizados e foi extremamente gratificante, apesar de muito trabalhoso, criar materiais de raiz adequados aos alunos que encontrei.

A turma estava dividida em dois turnos, tinha dois professores diferentes, e por uma questão de equidade, os dois turnos devem aprender os mesmos conteúdos, seguindo as mesmas estratégias e metodologias. Embora isto me levante algumas dúvidas, muitas vezes as estratégias e atividades aplicadas a um turno podem nem sempre funcionar no outro, pois estamos a falar de alunos e professores diferentes. A verdade é que com a devida antecedência tinha de ceder os meus materiais ao colega, informando-o das atividades e metodologias que ia utilizar, para que ele as colocasse em prática no seu turno. Para tal decidi criar uma conta partilhada na *DropBox* (figura 36), onde coloquei

por aula, todos os materiais educativos com as respetivas soluções, e um documento onde descrevi a sequência de utilização dos mesmos. No entanto, como eu ia recolhendo dados frequentemente, a minha ação estava constantemente a ser reformulada, o que me obrigava a gerir a entrega dos materiais da melhor maneira possível.

O conjunto destes fatores, associados ao facto de a disciplina ter tido uma carga horária semanal de 6 blocos de 90 minutos distribuídos por 3 dias seguidos, aumentou a complexidade de implementação do projeto, no entanto, encarei estas dificuldades como um novo desafio e penso que o superei com sucesso.

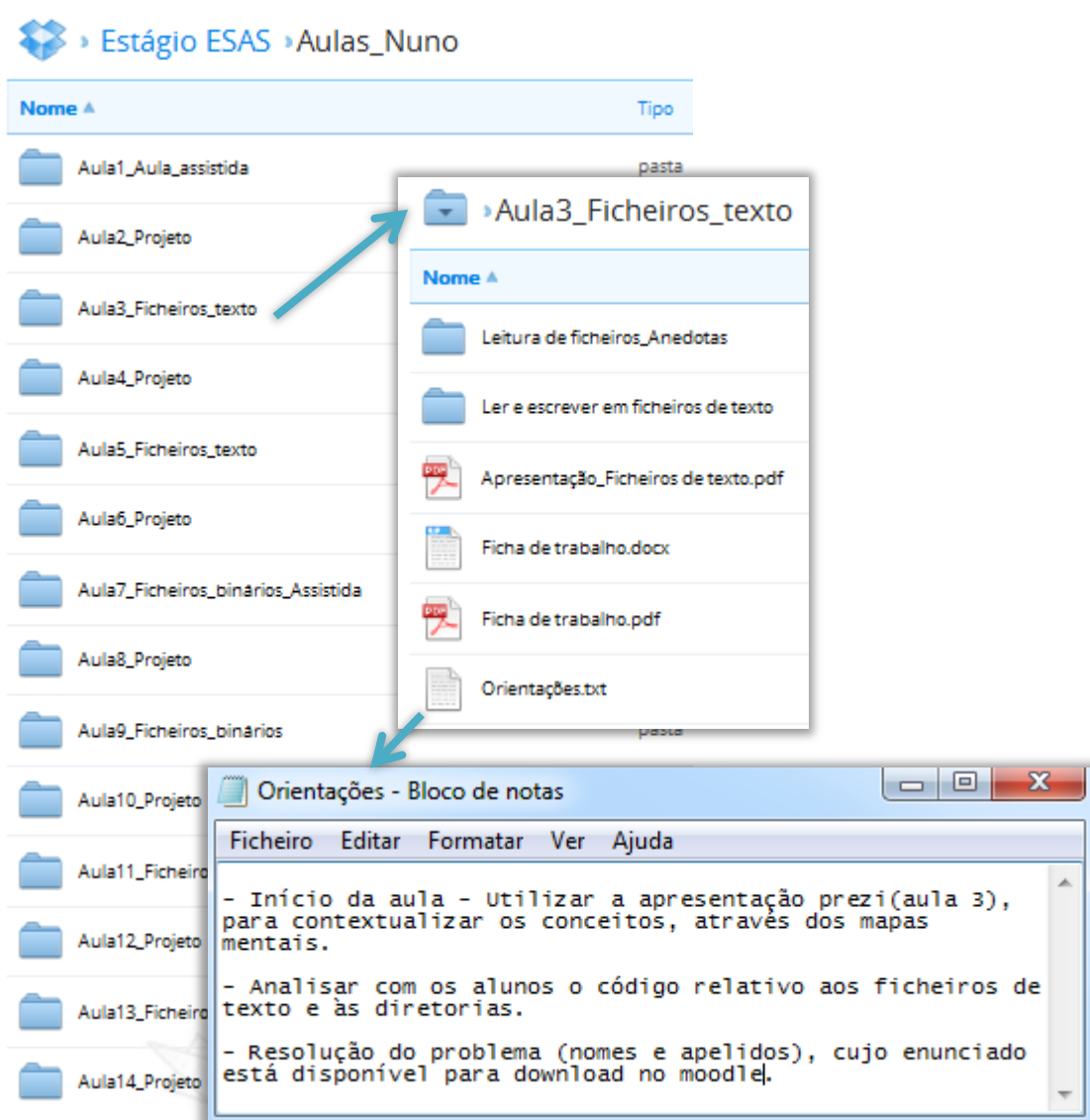


Figura 36 - Esquematização da partilha de dados com professor responsável pelo outro turno

CONCLUSÕES

Este trabalho teve como princípio a busca de boas práticas para o ensino da Programação Orientada a Objetos no ensino profissional. A minha experiência profissional de 5 anos a lecionar programação em cursos profissionais, utilizando metodologias baseadas na exposição e na instrução direta, permitiu-me estabelecer a comparação com esta abordagem construtivista e confirmar que esta se adequa melhor ao público-alvo e ao ensino da programação em cursos profissionais.

Um ambiente de aprendizagem construtivista permite uma maior destreza e envolvimento dos alunos dotando-os de competências úteis a longo prazo. Este ponto é de extrema riqueza, na medida em que possibilita aos alunos a oportunidade de desenvolverem habilidades, como a capacidade de interpretar problemas, de formular respostas, de saber pesquisar e filtrar informação e de desenvolver o raciocínio abstrato. Levar os alunos a adquirir estas competências é tão importante ou mais do que ensinar conteúdos. Invocando aqui uma célebre frase de Richard Riley⁴, Secretário da Educação durante a presidência do Bill Clinton, na qual refere que atualmente temos de preparar os alunos para empregos que ainda não existem, tecnologias que ainda não foram inventadas e problemas que ainda não são problemas. Este pensamento demonstra a importância do que expus anteriormente e torna-se ainda mais pertinente por falarmos da informática, onde a evolução não para e por isso é tão importante aprender a aprender.

Um outro aspeto que, na minha opinião, distingue as abordagens de ensino tradicionais das baseadas em pressupostos construtivistas, é a perceção que os alunos têm da sua aprendizagem. Em metodologias tradicionais, o aluno no seu estado passivo, ao ouvir ou ver fazer o professor, fica com a perceção, muitas vezes errada, que conseguiu assimilar o conhecimento quando na verdade não conseguiu. Isto porque muitas vezes as coisas parecem triviais quando vemos o outro fazer. Já em metodologias construtivistas o aluno, através do seu envolvimento e participação ativa no processo de ensino-aprendizagem, tem a oportunidade de testar os seus conhecimentos e assim aperceber-se das suas dificuldades.

Outro ponto de divergência entre as metodologias está no ritmo de aprendizagem dos alunos, a abordagem construtivista, através da resolução de problemas e projetos, permite aos alunos evoluírem a ritmos diferentes. Alunos com mais destreza avançam mais depressa e tornam as suas soluções mais elaboradas, e alunos com mais dificuldades, avançam mais devagar e tornam as suas soluções

⁴ Citado por Karl Fish - Did You Know/Shift Happens – video 2010

mais simples. Em abordagens mais tradicionais, existe a tendência para o professor impor um ritmo de aprendizagem igual para todos, o que me parece ser um gerador de desmotivação para alguns alunos. O que normalmente acontece é a definição de um nível de exigência, que permite à maioria progredir ao mesmo ritmo, mas aplicando o conceito da curva de Gauss, vão existir dois extremos para os quais este ritmo não é o adequado. Um dos extremos é constituído por alunos que não conseguem acompanhar o ritmo, e para quem o nível de dificuldade é elevado e fator de desmotivação. Já o outro é constituído por alunos, para quem o ritmo está muito lento, e o nível de dificuldade está muito baixo sendo fatores de desinteresse e dispersão.

Ao utilizar a abordagem construtivista, a diferença de ritmos de aprendizagem verificou-se ao longo de todo o projeto e foi comprovada no final quando nem todos terminaram a sua solução ao mesmo tempo.

Foi fundamental, para o sucesso da utilização desta abordagem pedagógica, conhecer as diversas metodologias associadas ao construtivismo, para assim fazer a sua gestão e aplicá-las em função das necessidades de aprendizagem.

Convém referir que apesar das inúmeras vantagens do construtivismo, não concordo com o construtivismo radical, aprendi ao longo desta intervenção que também devem existir momentos onde o professor ensina os conceitos. Por um lado, porque também nesta abordagem pedagógica é preciso fornecer alguns conteúdos aos alunos para que eles possam trabalhar sobre eles, e por outro lado, porque no ensino de conceitos a relação entre o tempo gasto e os conceitos adquiridos é desfavorável. Esta intervenção foi maioritariamente construtivista, pois, segundo a minha experiência, não vamos voltar a reconstruir tudo do zero, sendo assim, alguns conceitos foram ensinados de forma mais tradicional, mais expositiva, e as interligações e a própria flexibilidade cognitiva de forma construtivista, porque a compreensão, esta sim, é algo que só eles constroem individualmente, e não algo que se transmite de professor para aluno. Desta forma, inicialmente os conceitos eram apresentados alternando entre o expositivo e metodologias mais práticas como a resolução de exercícios, análise de mapas mentais e o uso de analogias para facilitar a sua compreensão, seguidamente estes conceitos eram postos em prática e relacionados com outros através das metodologias de aprendizagem baseada em projetos e problemas.

Relativamente à relação entre as metodologias e a tecnologia, verifiquei que a abordagem construtivista, ao estimular a participação ativa dos alunos, permitiu uma melhor e maior interação com as tecnologias.

As diferentes tecnologias tiveram o seu papel neste projeto e foram recursos essenciais no sucesso do mesmo, permitindo e dando suporte à construção de aprendizagens. Ao optar por escolher tecnologias através da técnica de intersecção, já explicada neste relatório, previ, logo à partida, que elas acrescentariam valor ao ambiente de aprendizagem, e se adequariam à metodologia e necessidades de aprendizagem dos alunos. Criou-se assim a conexão perfeita entre os ideais construtivistas e as características técnicas da tecnologia utilizada.

Perkins (1992) classifica como construtivista um ambiente de aprendizagem que ofereça ao aluno ferramentas de construção e a possibilidade de interação com a realidade, muitas vezes simulada. A plataforma Microsoft Visual C# possibilitou ao aluno vivenciar a construção de aprendizagens num ambiente em tudo semelhante ao real, por um lado porque permitiu a concretização de problemas reais e por outro, porque é uma ferramenta utilizada em muitas empresas.

A postura perante o professor e o relacionamento que o aluno tem com o mesmo, contribuem em grande parte para o sucesso da aprendizagem e para a motivação extrínseca, incutindo-lhe a vontade de aprender e de prolongar o esforço, desistindo menos, pois a perceção de competência que o aluno tem relativamente à sua aprendizagem pode ser desmotivante, caso ele não receba do professor um feedback positivo, que demonstre o esforço empreendido. O professor deve demonstrar que está atento ao aluno e acompanha o seu trabalho, incentivando-o, referindo, por exemplo, “ muito bem, gostei dessa parte que modificaste” ou “por acaso essa função foi uma boa ideia”. Esta atenção e acompanhamento dados ao trabalho dos alunos motiva-os e leva-os a continuar a trabalhar, procurando o reconhecimento.

Eu sempre pensei que ser professor de informática era bastante exigente, tanto pelo facto de ser uma área em constante evolução, obrigando a uma atualização permanente, como por os alunos poderem apresentar uma panóplia de dúvidas e dificuldades inerentes à tecnologia, à sintaxe da programação e aos erros de algoritmia. Agora, com a experiência adquirida neste projeto, constatei que ainda se torna mais exigente quando optamos por utilizar abordagens construtivistas, onde os alunos fazendo uso da sua liberdade, seguem por caminhos desconhecidos, e cabe ao professor ter a capacidade e a destreza de os orientar por esses caminhos. Existe sempre uma imprevisibilidade do rumo da aula, estando este sujeito a constantes mudanças em detrimento das soluções e interesses que os alunos achem mais apropriados. Este facto impõe ao professor o dever de estar muito bem preparado, científica e pedagogicamente, para que possa cumprir o seu papel de uma forma irrepreensível, porque só assim conseguirá sair da sua zona de conforto, abandonar o seu guião e entrar na zona de conforto dos alunos.

Para além deste esforço acrescido dentro da sala de aula, o professor ainda tem de se deparar com o desafio de planejar atividades e estratégias adequadas, que permitam aos alunos construir o seu conhecimento.

Mesmo assim, penso que foi uma aposta conseguida, na medida em que compensou pelos resultados obtidos. É triste que muitos professores não sejam sensíveis à mudança, por razões como o enraizamento das metodologias utilizadas, resistência, medo ou a falta de preparação para a mesma, embora compreenda que muitos não o façam pela falta de tempo resultante da sobrecarga de trabalho a que têm sido sujeitos. Mas mais lamentável ainda são aqueles que dizem aplicar uma abordagem construtivista, quando na verdade o que estão a fazer é simplificar o seu trabalho, apresentando problemas e projetos aos alunos, mas ignorando por completo a proximidade orientativa.

Há fatores externos à escola sobre os quais, nós enquanto professores, não temos controlo, mas que têm um grande impacto no rendimento dos alunos. Como um dos mais importantes, saliento os ambientes familiares deteriorados e pais negligentes. Esta situação implica, na maioria dos casos, o não reconhecimento de regras pelos alunos, quer na escola, quer em casa. As duas situações são prejudiciais para os alunos, na escola, o aluno não aceita as recomendações do professor e não consegue ser disciplinado no processo de aprendizagem. Em casa o aluno tem tendência para não estudar e muitas vezes não dorme nem come em condições. Estes elementos vão-se refletir negativamente nas suas aprendizagens e na sua concentração.

O presente estudo foi realizado num contexto específico, para esta turma em particular. Noutro contexto, algumas metodologias e principalmente, os materiais utilizados, devem ser adaptados aos alunos em questão.

Uma das limitações do estudo apresentado prende-se essencialmente com o tamanho da amostra. E como tal, uma sugestão futura de investigação passa por alargar este estudo a outras turmas e a outras disciplinas de informática, permitindo assim validar o presente estudo.

Apesar da complexidade dos conteúdos, dos alunos e do contexto em si, com as metodologias utilizadas, associadas às tecnologias adequadas e com o acompanhamento apropriado para esta turma, consegui mudar comportamentos e ensinar os alunos a aprender a aprender e a aprender a fazer, solucionando-se assim problemas atuais e anteriores ao nível da aprendizagem da programação, autonomia, motivação e comportamento. Esta abordagem, ao permitir que os alunos se empenhem e se envolvam nas suas aprendizagens, acaba por minimizar os problemas de comportamento.

BALANÇO FINAL DA FORMAÇÃO

Cheguei finalmente ao ponto em que é necessário fazer um balanço do meu desempenho ao longo deste projeto de intervenção. Apresentarei apenas algumas conclusões, uma vez que já se enunciaram tantas ao longo deste percurso. Resta-me apenas concluir esta etapa, fazendo uma síntese das impressões, dificuldades e algumas observações gerais.

Este projeto de intervenção contribuiu muito para o desenvolvimento e aperfeiçoamento das minhas práticas profissionais sustentadas em investigação atualizada, na observação e na reflexão crítica. Sinto ser importante salientar a sua contribuição para o meu crescimento a nível científico, pois ao lecionar o último módulo de Programação Orientada a Objetos, tive primeiro que aprender coisas novas para depois poder ensinar uma vez que esta linguagem de programação não fez parte da minha formação base. Aprendi que é essencial para sermos professores competentes em qualquer área, termos um bom nível de conhecimentos científicos, para estarmos seguros e transmitirmos segurança aos alunos.

Enquanto promovia a autonomia dos alunos, sem me aperceber, no meio desta envolvimento toda, acabei por estimular também a minha própria autonomia, a minha confiança, a minha aprendizagem, melhorando no modo como encaro uma turma e as dificuldades que ela apresenta. Este estágio não só me enriqueceu a nível científico, mas também a nível pedagógico, pois dotou-me de ferramentas para desenvolver e proporcionar momentos onde os alunos foram protagonistas das suas aprendizagens, onde existiu espaço para o diálogo, colaboração, participação, mobilização de recursos e aprendizagens.

Este projeto foi muito importante para a minha formação enquanto professor na medida em que me permitiu pôr em prática a abordagem construtivista do ensino, a qual tem aspetos e metodologias que se revelaram bastante eficazes e que em muito contribuem para o sucesso do processo de ensino-aprendizagem da programação.

Considero bastante gratificante verificar como a tecnologia contribuiu para a implementação destas metodologias de ensino. A aplicação prezi ajudou a construir conhecimentos e a melhorar a participação; o moodle permitiu manter um repositório de todos os materiais organizados aula a aula; a Dropbox permitiu criar um e-portfolio para guardar os trabalhos e reflexões; o quadro interativo cativou a atenção dos alunos para a resolução de exercícios e a aplicação Microsoft Visual Studio C#

possibilitou aos alunos colocarem em prática tudo o que aprenderam sobre programação, recebendo um feedback imediato.

O estágio não se resumiu apenas a um momento de prática, mas sim a uma articulação e mobilização de diversas aprendizagens construídas ao longo do curso. Nesse sentido, tendo por base as aprendizagens transmitidas no mestrado, fui refletindo sobre a minha prática, sempre na tentativa de encontrar novas estratégias para que o meu trabalho pudesse proporcionar condições de aprendizagem significativas para a turma. Acredito que os professores conseguem mudar as suas práticas refletindo, só uma reflexão da ação nos permite reavaliar o nosso trabalho e procurar possíveis soluções.

Com esta reflexão cuidada, consegui definir os objetivos, as atividades e as metodologias de acordo com as capacidades, dificuldades e conhecimentos dos alunos, promovendo assim uma participação ativa dos mesmos.

Aprendi que a observação é muito importante, só através dela é que consegui detetar as dificuldades que os alunos tinham. E foi a observação que me conduziu a aspetos não menos importantes que são a reflexão e a investigação, as quais me permitiram encontrar soluções para mudar a minha ação, no sentido de melhorar todo o processo de ensino-aprendizagem e assim colmatar os problemas encontrados.

A verdade é que toda esta experiência me tornou um professor mais competente, seguro de mim mesmo, investigativo, observador e reflexivo, e deixou-me consciente que devo continuar a apostar na minha formação, a nível científico e pedagógico, pois só assim cumprirei da melhor forma possível o meu papel como professor.

Futuramente pretendo continuar a aperfeiçoar e aplicar as estratégias e atividades concretizadas ao longo deste projeto e que, com sucesso, contribuirão para um ensino melhor, mais integrador e próximo das necessidades e dificuldades dos alunos.

A nível pessoal, tornei-me um homem diferente. Encontrei uma turma muito heterogénea, com histórias pessoais muito difíceis e não consegui ser apenas o professor. Estes alunos evoluíram não só a nível de desempenho escolar, mas também como pessoas. No pouco tempo que estive em contacto com eles, conseguimos estabelecer uma relação de companheirismo e confiança. Eles viam-me como professor e sempre respeitaram a minha autoridade, apesar de ser apenas um professor “estagiário”, mas tinham abertura para vir ter comigo se precisassem de conversar sobre algum assunto que os preocupasse.

Foi uma caminhada muito difícil, deparei-me com uma carga horária pesada, conteúdos que implicavam bastante preparação, alterações de calendário em cima da hora, uma turma complicada entre outros pormenores já referidos anteriormente, no entanto, estas dificuldades e obstáculos converteram-se em aprendizagem e tornaram o resultado final mais gratificante e enriquecedor.

REFERÊNCIAS BIBLIOGRÁFICAS

- AUSUBEL, D. (1978). *In defence of advance organisers: A reply to my critics*. Review of Educational Research, vol. 48(2), pp.251-257.
- ALCARÁ, A, GUIMARÃES, S. (2007). *A Instrumentalidade como uma estratégia motivacional*. *Psicologia Escolar Educacional*, vol.11 (1), pp.177-178.
- ANQEP (2013). *Gerações do Ensino Profissional*. Lisboa: Agência Nacional para a Qualificação e o Ensino Profissional.
- ATKINSON, J. (1964). *An Introduction to Motivation*. Princeton, N.J.: Van Nostrand.
- BANG, J. (2000). *A integração dos alunos imigrantes no sistema de ensino dinamarquês*. União Europeia. Luxemburgo: Cedefop.
- BOEKAERTS, M. (2003). *Motivation to learning*. New Delhi: Discovery.
- BRITO, M., SÁ-SOARES, F. (2010). *Computer Programming: Fail Fast to Learn Sooner*. Tech-Education. Sprinter-Verlag Berlin Heidelberg, pp. 223-229.
- BROOKS, J., BROOKS, M. (1997). *Construtivismo em sala de aula*. Porto Alegre: Artes Médicas.
- CARVALHO, Ana (2011), *A teoria da Flexibilidade Cognitiva e o Modelo Múltiplas Perspetivas*. CIEd.
- COLL, C., MARTIN, E., MAURI, T. MIRAS, M., ONRUBIA, J., SOLÉ, I., & ZABALA, A. (2001). *O construtivismo na sala de aula: Novas perspectivas para a acção pedagógica*. Porto: Edições ASA.
- COOPER A. (1993). *Paradigm Shifts in Design Instruction: From Behaviorism to Cognitivism to Constructivism*. Educational technology, vol. 33 (5), pp. 12-19.
- DETERDING, S., SICART, M., NACKE, L., O'Hara, K. & DIXON, D. (2011). *Gamification: Using game-design elements in nongaming contexts*. CHI '11 CHI Conference on Human Factors in Computing Systems. pp. 2425-2428.
- DICKINSON, L. (1987). *Self-instruction in language learning*. Cambridge: Cambridge University Press.
- DIRECÇÃO-GERAL DE FORMAÇÃO VOCACIONAL (2005). *Programa Componente da Formação Técnica Disciplina de Programação e Sistemas de Informação*. Disponível em www.anqep.gov.pt e consultado em 22 de novembro de 2012.

- DONNELLY, R., FITZMAURICE, M. (2005). *Collaborative project-based learning and problem-based learning in higher education: a consideration of tutor and student roles in learner-focused strategies*. In O'Neill, G., Moore, S., McMullin, B. (eds): *Emerging Issues in the Practice of University Learning and Teaching*. Dublin: AISHE.
- ESAS. (2011). *Escola Secundária Alberto de Sampaio - Projeto Educativo*. Disponível em www.esas.pt e consultado em 25 de novembro de 2012.
- FERRAZ, D., TERRAZZAN, E. (2002). *O uso espontâneo de analogias por professores de Biologia: observações da prática pedagógica*. Ensaio Pesquisa em Educação em Ciências, vol. 4 (2), pp. 1-15.
- FREIRE, P. (1997). *Pedagogia da autonomia*. São Paulo: Paz e Terra.
- GAVA, T., MENEZES, C., & CURY, D. (2003). *Aplicações de Mapas Conceituais na Educação como Ferramenta MetaCognitiva*. III International Conference on Engineering and Computer Education - ICECE 2003, São Paulo.
- Harrison, A., Treagust, D. (2006). *Teaching and learning with analogies*. In Aubusson, P., Harrison, A., Ritchie, S. (Eds.), *Metaphor and analogy in science education* Dordrecht, The Netherlands: Springer, vol. 30, pp. 11–24.
- IGEC. (2011). *Avaliação Externa das Escolas - Relatório Escola Secundária de Alberto Sampaio*. Disponível em www.ige.min-edu.pt e consultado em 25 de novembro de 2012.
- JESUS, S. (1997). *Influência do professor sobre os alunos*. Porto: Edições ASA
- JIMÉNEZ R., M., LAMB, T. & VIEIRA, F. (2007). *Pedagogia Para a Autonomia na Educação em Línguas na Europa: para um quadro de referência do desenvolvimento do aluno e do professor*. Dublin: Authentik.
- JOÃO, M.. (2003). *Programa de Tecnologias da Informação e Comunicação: 9º e 10º Anos*. Ministério da Educação: Direção-Geral da Inovação e Desenvolvimento Curricular.
- JONASSEN, M. D. (1991). *Objectivism versus Constructivism: Do We Need a New Philosophical Paradigm?*. Educational Technology Research and Development, vol. 39(3), pp. 5-14.
- KORDAKI, M. (2010). *A drawing and multi-representational computer environment for beginners' learning of programming using C. Design and pilot formative evaluation*. Computers & Education, vol. 54(1), pp.69-87.
- LOURENÇO, A., PAIVA, M. (2010). *A motivação escolar e o processo de aprendizagem*. Ciências & Cognição, Vol. 15(2), pp.132-141.
- MACEDO, L. (1994). *Ensaio Construtivistas*. São Paulo: Casa do Psicólogo.

- MARQUES, P., PEDROSO, H. & FIGUEIRA, R. (2011). *C# 4.0*. Lisboa: FCA.
- MED (2010). *Guia / conselhos. Programação C*. Ministério da Educação da Dinamarca. Disponível em www.uvm.dk e consultado em 14 de novembro de 2012.
- MULLEN, B., JONHSON, C. & SALAS, E. (1991). *Productivity loss in Brainstorming Groups: a meta-analytic integration*. Basic and Applied Social Psychology, vol. 12 (1), pp. 3-23.
- PERKINS, N. (1992). *Technology Meets Constructivism: Do They Make a Marriage?* In: DUFFY, T.M., JONASSEN, D.H. (Eds.). Constructivism and the Technology of Instruction: A Conversation. NJ: Lawrence Erlbaum.
- PIAGET, J. W. (1969). *Psicologia e Pedagogia*. (D. A. Lindoso, & R. M. Silva, Trans.) Rio de Janeiro: Forense Universitária.
- PIAGET, J. W. (1979). *A construção do real na criança*. (Â. Cabral, Trans.) Rio de Janeiro: Zahar.
- SHUNK, D. (1991). *Self-Efficacy and Academic Motivation*. Educational Psychologist, vol. 26 (3&4), pp. 207-231.
- SPIRO, R., JEHNG, J. (1990). Cognitive flexibility and hypertext: Theory and technology for the non-linear and multi-dimensional traversal of complex subject matter. In D. Nix and R. J. Spiro (Eds.), Cognition, education, and multimedia: Exploration in high technology. Hillsdale, NJ: Lawrence Erlbaum.
- SPIRO, R., VISPOEL, W., SCHMITZ, J., SAMARAPUNGAVAN A. & BOERGER, A. (1987). *Knowledge Acquisition for Application: Cognitive Flexibility and Transfer in Complex Content Domains*. Technical Report N° 409. University of Illinois at Urbana-Champaign.
- SPIRO, R., COULSON R., FELTOVICH & P., ANDERSON D. (1988). *Cognitive Flexibility Theory: Advanced Knowledge Acquisition in Ill-Structured Domains*. In Tenth Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Erlbaum, pp. 375-383.
- THOMAS, J. (2000). *A review of research on project-based learning*. (CA:Autodesk, Ed.). Disponível em www.K12reform.org/foundation/pbl/research e consultado em 17 de Janeiro de 2013.
- WATTS, M. (1991). *The science of problema-solving*. Londres: Cassell Education.

ANEXOS

Anexo 1: Planificação do módulo 11

Módulo n.º	11	Designação:	Programação Orientada a Objetos Avançada		
Objetivos de ensino/ Conteúdos		Objetivos de Aprendizagem/Competências	Estratégias/Sugestões Metodológicas	Duração (Blocos de 90 min.)	
<ul style="list-style-type: none">Fazer o tratamento de erros de uma maneira estruturada.Gestão de erros e exceções com C#Virtualizar fluxos de dados através do conceito de Stream.Manipulação de Streams em diversos contextosLeitura e escrita de ficheiros (texto e binários) em C#Estruturar soluções usando o paradigma da programação orientada a objetos.		<ul style="list-style-type: none">Introdução ao conceito de erro e Exceção.Manipulação de Exceções.Criação de Exceções próprias.Introdução ao conceito de Stream.Derivação de StreamsFicheiros de texto e Binários	<ul style="list-style-type: none">O Professor deve introduzir os conteúdos de uma forma clara, recorrendo a exemplos concretos e reais.O professor deve confrontar os alunos com um conjunto de problemas de complexidade crescente com o objetivo de consolidar os conceitos desenvolvidos anteriormenteIncentivar os alunos a participarem na realização de miniprojectos, para a resolução de problemas e em exercícios que simulem a realidade.	25	
Recursos:	Computadores, Internet, Microsoft Visual C#, Projetor de Vídeo, Manuais produzidos pelos professores				
Avaliação:	A avaliação deste módulo será efetuada com base em: Atitudes e Valores: 20%; Cognitivo: 80%.				

Anexo 2: Guião da tarefa 1

Tarefa 1 – Criação da classe

Tempo aproximado para completar a tarefa: 60 minutos

Objetivos:

- Definir variáveis
- Implementar construtores
- Encapsular os dados

Visão geral da tarefa:

Criar uma classe que agrupe todas as variáveis necessárias para os objetos. Deve encapsular os dados e implementar os construtores que achar necessários.

Ajuda à tarefa (sugestão):

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Projeto_final
{
    public class exemplo
    {
        private int variavel_exemplo1;
        private string variavel_exemplo2;

        public exemplo ()
        {

        }

        public jogador(int variavel_1, string variavel_2)
        {
            variavel_exemplo1 = variavel1;
            variavel_exemplo2 = variavel2;
        }

        public void set_variavel1 (int variavel1)
        {
            variavel_exemplo1 = variavel1;
        }
    }
}
```



```
public void set_variavel2(string variavel2)
{
    variável_exemplo2 = variavel2;
}

public int get_variavel1()
{
    return variável_exemplo1;
}
public string get_variavel2()
{
    return variável_exemplo2;
}
}
}
```

Anexo 3: Guião da tarefa 2

Tarefa 2 – Criação do formulário de login

Tempo aproximado para completar a tarefa: 180 minutos

Objetivos:

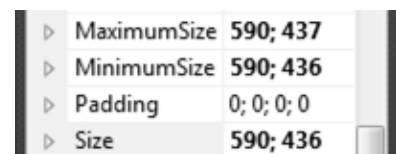
- Criar formulários
- Alterar propriedades de formulários.
- Inserir menustrip
- Criar diretorias
- Criar ficheiros de texto

Visão geral da tarefa:

Criar um formulário para fazer login na aplicação, os dados do login devem ser guardados num ficheiro e este dentro de uma diretoria. A abertura dos formulários deve ser controlada por um menu.

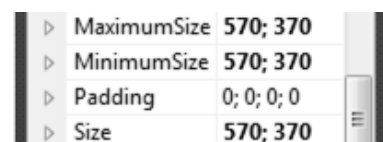
Instruções da tarefa (sugestões):

1. No formulário principal insira um objeto menuStrip
2. Insira um item no menu “Inserir elemento”
3. Altere o formulário para MDI
4. Altere a propriedade `MaximizeBox` para `false`
5. Altere as dimensões do formulário para



▷ MaximumSize	590; 437
▷ MinimumSize	590; 436
▷ Padding	0; 0; 0; 0
▷ Size	590; 436

6. Crie um novo formulário para inserir elementos com o nome `Form_inserir`.
7. Remova as bordas do formulário alterando a propriedade `FormBorderStyle` para `none`
8. Altere as dimensões do formulário para



▷ MaximumSize	570; 370
▷ MinimumSize	570; 370
▷ Padding	0; 0; 0; 0
▷ Size	570; 370

9. Associe este novo formulário ao item do `menuStrip` “inserir elemento”
10. Crie uma diretoria com o nome “Diretoria” no método `load` do formulário principal.
11. Tem de testar se a diretoria já existe para evitar estar sempre a criá-la e apagar o seu conteúdo cada vez que entramos na aplicação
12. Crie um novo formulário para login com o nome `form_login`
13. Altere as propriedades do formulário como alterou para o formulário `inserir`.
14. Insira no formulário duas `labels` e duas `textBoxs` para o utilizador inserir os dados do login
15. Insira um botão para validar o login, verificar se os dados introduzidos estão dentro do ficheiro.

16. Insira um botão no menuStrip (no form principal) para abrir o formulário login (ver Ajuda à tarefa).
17. Crie um ficheiro de texto para guardar os dados do login, no método load do formulário principal.
18. O botão login deve verificar se os dados introduzidos pelo utilizador existem no ficheiro, se existirem deve desbloquear o menuStrip.
19. Quando o login for efetuado com sucesso deve fechar este formulário. (this.close)

Ajuda à tarefa (sugestões):

- Código para abrir um formulário verificando primeiro se ele já estava aberto impedindo assim a abertura duplicada

```
private void inserirToolStripMenuItem_Click(object sender, EventArgs e)
```

```
{
    // Instanciar o novo formulário
    Form_inserir form_novo = new Form_inserir();

    // Verificar se já está aberto, para n abrir duplicado
    foreach (Form formFilho in this.MdiChildren)
    {
        if (formFilho.GetType() == form_novo.GetType())
        {
            formFilho.Focus();
            return;
        }
    }

    form_novo.MdiParent = this;
    form_novo.Show();
}
```

Notas:

- Menustrip tem de estar enabled antes de se efetuar o login
- Tem de se fechar sempre os streams para se poder escrever nos ficheiros.
-

Solução sugerida:

Form1

Inserir jogador

login

login:

Password:

Entrar

Anexo 4: Guião da tarefa 3

Tarefa 3 – Manipulação de Dicionário e GridView

Tempo aproximado para completar a tarefa: 180 minutos

Objetivos:

- Manipular a coleção Dictionary
- Manipular o objeto GridView

Visão geral da tarefa:

Utilizar a classe Dictionary para criar uma coleção dos objetos, a aplicação deve ser capaz de inserir dados na coleção e listar os dados da coleção.

A aplicação deve permitir listar todos os dados da coleção numa ListView.

Anexo 5: Guião da tarefa 4

Tarefa 4 – Manipulação de Ficheiros Binários e Serialização de Classes

Tempo aproximado para completar a tarefa: 180 minutos

Objetivos:

- Manipular ficheiros binários
- Aprender a serializar classes
- Manipular os objetos de formulários
- Remover elementos de uma listview
- Remover elementos de um dicionário

Visão geral da tarefa:

Nesta tarefa deverá criar um ficheiro binário para guardar os dados da coleção dictionary. A classe dictionary deve ser serializada para facilitar a sua gravação em ficheiros.

Instruções da tarefa (sugestões):

1. Crie um ficheiro binário no formulário principal para guardar a coleção dictionary.
2. Defina a classe que criou na tarefa 1 como *serializable*
3. No Form_inserir no botão guardar, guarde o dicionário no ficheiro binário que abriu anteriormente.
4. No form_listar insira um botão para remover elementos da listview e do dicionário.
5. Altere as textBoxes do formulário inserir por controlos mais amigáveis ao utilizador.

Ajuda à tarefa (sugestões):

- Exemplo: serializar uma classe
namespace Projeto_final

{ [Serializable]

Anexo 6: Diário de aprendizagem



Escola Secundária de Alberto Sampaio

Ano Letivo
2012-2013

Curso:	Técnico de Gestão e Programação de Sistemas Informáticos	Ano/ Turma:	2P
Disciplina:	Programação e Sistemas de Informação		

Nome	Número	Data
------	--------	------

Diário de Aprendizagem

<i>O que aprendi hoje?</i>
<i>O que fiz hoje?</i>
<i>As dificuldades que encontrei e como as resolvi?</i>

Anexo 7: Grelha de avaliação do projeto desenvolvido pelos alunos



GOVERNO DE
PORTUGAL

MINISTÉRIO DA EDUCAÇÃO
E CIÊNCIA

Escola Secundária de Alberto Sampaio 2012/13

Ano/Turma	2P	Disciplina	PSI	Curso	Técnico de Gestão e Programação de Sistemas informáticos
-----------	----	------------	-----	-------	--

Nº	Nome	Foto	Apresentação					Tarefa1	Tarefa2	Tarefa 3		Tarefa 4			Total	Observações
			Postura	Conhecimento sobre a aplicação.	Empenho	Autonomia	Criat/	Classe	Login	Dicti/	ListView	Remover elementos	Serialização	Controlos		
			15	15	20	15	15	15	20	15	20	10	20	20	200	
15															0	
16															0	
17															0	
20															0	
21															0	
22															0	
23															0	
24															0	
25															0	
26															0	
28															0	
29															0	

0 Total de Alunos



UNIÃO EUROPEIA

Fundo Social Europeu



Governo da República Portuguesa

Anexo 8: Grelha de observação de comportamentos



GOVERNO DE
PORTUGAL

MINISTÉRIO DA EDUCAÇÃO
E CIÊNCIA

Escola Secundária de Alberto Sampaio 2012/13

Grelha de Observação

Ano/Turma			2P		Disciplina	PSI	Curso	Tec. de Gestão e Programação de Sistemas informáticos				
Nº	Nome	Foto	Assiduidade	Pontualidade	Postura	Material	Interesse	Participação	Autonomia	Motivação	Total	Observações
			30%	50%	20%	10%	15%	15%	20%	20%	200	
15												
16												
17												
20												
21												
22												
23												
24												
25												
26												
28												
29												
0 Total de Alunos			(C) Concretizou		(CR) Concretizou Razoavelmente		(NC) Não concretizou					



QUADRO
DE REFERÊNCIA
ESTRATÉGICO
NACIONAL

UNIÃO EUROPEIA

Fundo Social Europeu



Governo da República Portuguesa

Anexo 9: Questionário de autoavaliação dos alunos e avaliação do projeto de intervenção pedagógica.

Autoavaliação

*Obrigatório

Nome *

Número *

Autoavaliação *

	Sempre	Muitas vezes	Às vezes	Nunca
Cumpri as regras de comportamento estabelecidas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Interagi adequadamente com o(s) colega(s).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Contribuí para o enriquecimento da aula (fiz sugestões e/ou críticas construtivas).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Realizei as tarefas que me foram indicadas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fui autônomo/a na gestão das atividades.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Senti necessidade de pedir ajuda e/ou esclarecimentos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Geri adequadamente o meu tempo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
As atividades agradaram-me.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fui pontual.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Avaliação do projeto *

	Sempre	Muitas vezes	Às vezes	Nunca
Gostei da forma como as aulas foram lecionadas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gostei das atividades propostas pelo professor.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O professor tirou as minhas dúvidas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Indica as coisas que mais gostaste nestas aulas. *

Indica as coisas que menos gostaste.

Enviar

Nunca envie palavras-passe através dos Formulários Google.

Com tecnologia
Google Drive

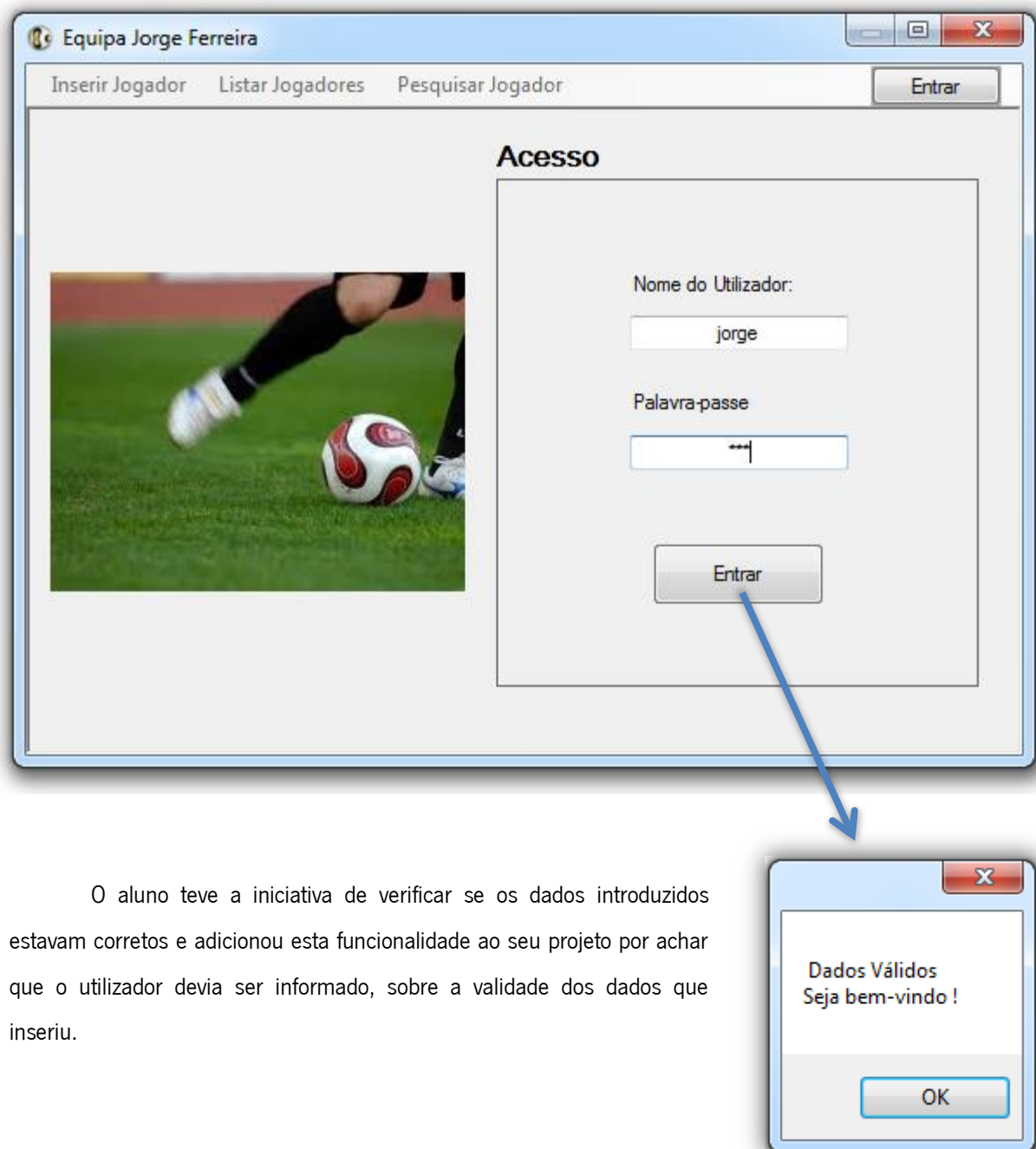
Este conteúdo não foi criado nem aprovado pela Google.
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)

Anexo 10: Exemplo de projeto desenvolvido pelos alunos

Este anexo demonstra algumas das soluções desenvolvidas pelos alunos e ao mesmo tempo funcionalidades e pormenores que os alunos adicionaram de forma autónoma.

1. Menu de login

O aluno desenvolveu o menu principal da aplicação com sucesso e todo o formulário de login. Este aluno decidiu adicionar um ícone à sua aplicação.



O aluno teve a iniciativa de verificar se os dados introduzidos estavam corretos e adicionou esta funcionalidade ao seu projeto por achar que o utilizador devia ser informado, sobre a validade dos dados que inseriu.

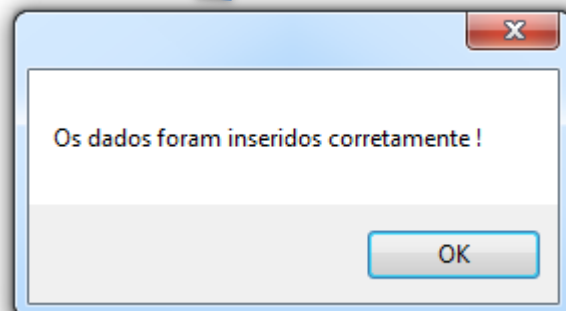
2. Inserir um novo jogador

Depois de efetuar o login correto o utilizador já pode inserir novos jogadores na sua aplicação. O aluno inseriu os controlos corretamente neste formulário, por forma a simplificar a inserção dos dados.



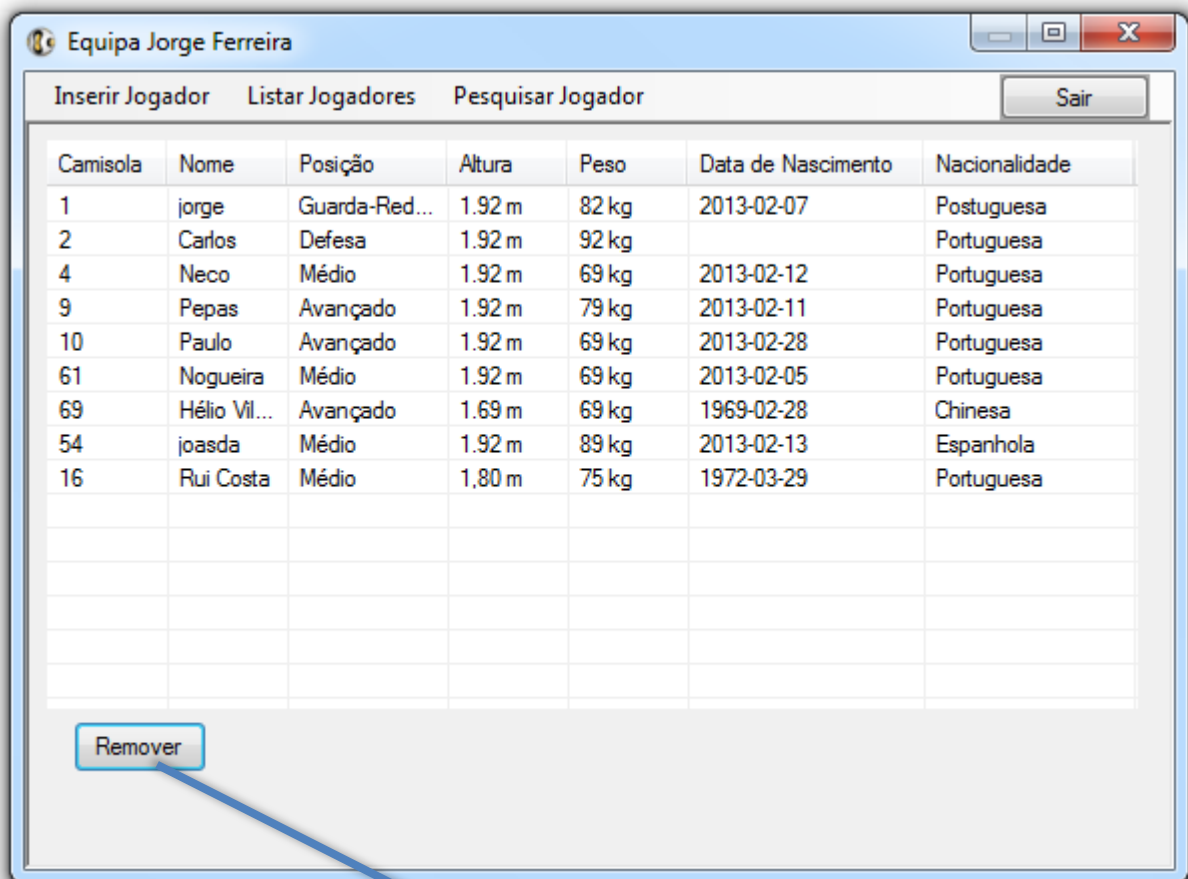
The screenshot shows a Windows-style application window titled "Equipa Jorge Ferreira". It has a menu bar with "Inserir Jogador", "Listar Jogadores", and "Pesquisar Jogador", and a "Sair" button. The main area is titled "Formulário para inserir Jogador". It contains several input fields: "Camisola" (16), "Nome futebolístico" (Rui Costa), "Data de Nascimento" (29-03-1972), "Altura" (1,80), "Peso" (75), and "Nacionalidade" (Portuguesa). On the right, there is a "Posição" section with radio buttons for "Guarda-Redes", "Defesa", "Médio" (selected), and "Avançado". Below this is a "Foto" section with a placeholder image of a footballer and a "Carregar Imagem" button. A "Guardar" button is at the bottom center. A blue arrow points from the "Guardar" button to a message box below.

O aluno verificou se os dados foram inseridos corretamente no ficheiro.

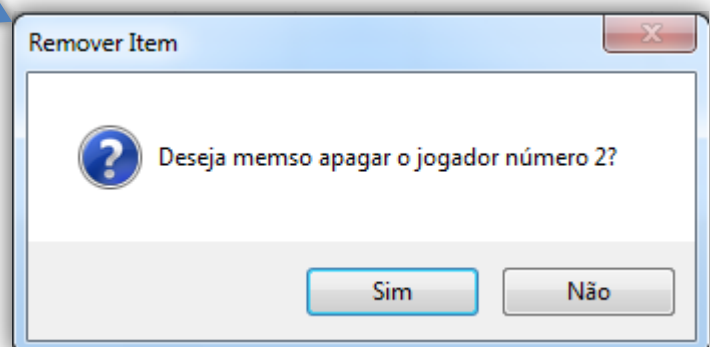


3. Listar jogadores

O aluno realizou com sucesso o formulário de listar jogadores.



Depois de selecionando um jogador é possível removê-lo.



4. Pesquisar jogadores

Por sua iniciativa o aluno achou que a sua aplicação precisava de um formulário para pesquisar jogadores. Assim decidiu criar o seguinte formulário, no qual o utilizador pode pesquisar jogadores em função do seu número de camisola.

The screenshot shows a Windows-style application window titled "Equipa Jorge Ferreira". It has a menu bar with "Inserir Jogador", "Listar Jogadores", and "Pesquisar Jogador", and a "Sair" button. The main area is titled "Formulário de Pesquisa de Jogador". It contains a form with the following fields and values:

Field	Value
Camisola	16
Nome futebolístico	Rui Costa
Data-de-Nascimento	1972-03-29
Altura	1,80 m
Peso	75 kg
Nacionalidade	Portuguesa
Posição	Médio

There is a "Fotografia" label above a photo of a player in a red and black jersey. Below the form is a "Limpar" button, and at the bottom center is a "Pesquisar" button.

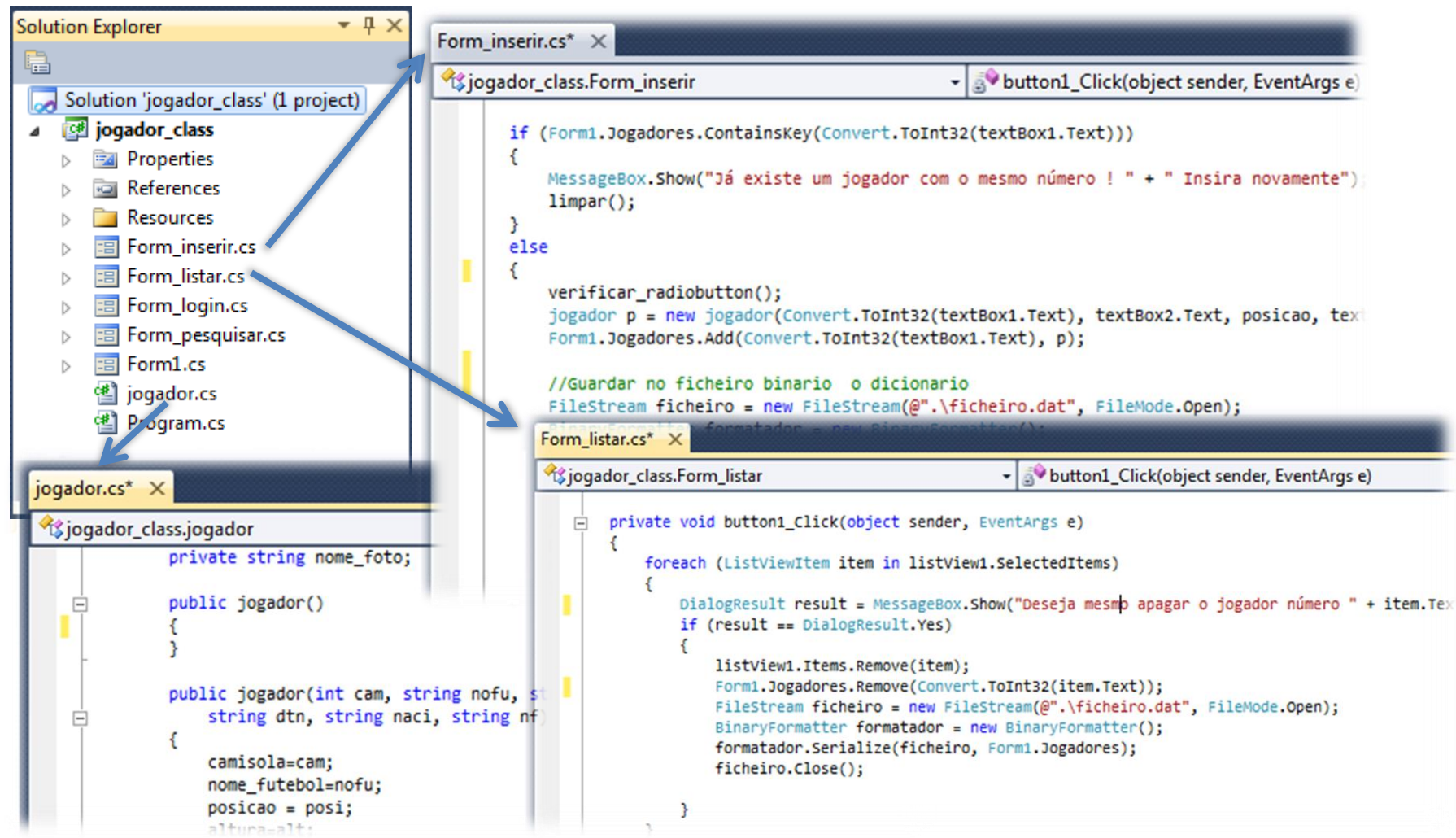
Outra solução

Um outro aluno na sua solução optou antes por colocar o pesquisar e o listar no mesmo formulário.
















Esta solução já permite ver a foto do jogador sempre que o selecionamos nas *listview* e também já permite encontrar um jogador na *listview* através do número da sua camisola.

[illegible]

Todas as classes e formulários criados tiveram de ser programados como se verifica nos seguintes exemplos.



Anexo 11: Relatório de visualizações por material na plataforma moodle.

Módulo 11 - Programação Orientada a Objetos Avançada			
 Questionário de Auto e Hétéro Avaliação	15	-	quinta-feira, 14 Fevereiro 2013, 13:43 (174 dias 5 horas)
 Apresentação das aulas - Prezi	169	-	segunda-feira, 11 Março 2013, 17:47 (149 dias 1 hora)
 Apresentação do projeto - Prezi	67	-	sábado, 16 Março 2013, 23:30 (143 dias 20 horas)
 Diário de Aprendizagem	33	-	terça-feira, 26 Fevereiro 2013, 11:31 (162 dias 7 horas)
 Submeter diário de aprendizagem	69	-	quinta-feira, 31 Janeiro 2013, 15:43 (188 dias 3 horas)
 Atividade 1	16	-	terça-feira, 26 Fevereiro 2013, 13:36 (162 dias 5 horas)
 Exercício 1	7	-	terça-feira, 19 Fevereiro 2013, 02:51 (169 dias 16 horas)
 Apresentação Projeto	26	-	terça-feira, 26 Fevereiro 2013, 10:56 (162 dias 8 horas)
 Tarefa 1	91	-	terça-feira, 19 Fevereiro 2013, 02:13 (169 dias 17 horas)
 Submeter diário de aprendizagem	48	-	quinta-feira, 31 Janeiro 2013, 15:43 (188 dias 3 horas)
 Exercícios ficheiros de texto	83	-	terça-feira, 19 Fevereiro 2013, 02:13 (169 dias 17 horas)
 Ficheiros com as anedotas	9	-	terça-feira, 19 Fevereiro 2013, 02:15 (169 dias 17 horas)
 Tarefa 2	116	-	terça-feira, 19 Fevereiro 2013, 02:15 (169 dias 17 horas)
 Exercício sobre a coleção Dictionary	14	-	terça-feira, 19 Fevereiro 2013, 02:16 (169 dias 17 horas)
 Tarefa 3	107	-	terça-feira, 19 Fevereiro 2013, 02:16 (169 dias 17 horas)